TRS-80 DATA COMMUNICATION SYSTEMS

A GUIDE TO THE OPERATION OF TRS-80 MICROCOMPUTERS AS COMMUNICATION DEVICES BY FRANK J. DERFLER, JR.

$12.95 A SPECTRUM BOOK

FRANK J. DERFLER, JR., a telecommunications manager for the federal government, has been actively involved in the development of microcomputer systems and their application to data communications since 1977. He is the author of *Microcomputer Data Communications Systems* (Prentice-Hall) and currently writes a monthly column about microcomputer-based data communicating systems for *Kilobaud Microcomputing* magazine.

Frank J. Derfler, Jr.

TRS-80 Data Communication Systems

A SPECTRUM BOOK

Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

Derfler, Frank J.
TRS-80 data communication systems.

Includes index.

1. Data transmission systems. 2. TRS-80 (Computer) I. Title. II. Title: T.R.S.-80 data communication systems.

TK5105.D483 001.64'4 82-3834
ISBN 0-13-931238-2 AACR2
ISBN 0-13-931220-X (pbk.)

This Spectrum Book is available to businesses and organizations at a special discount when ordered in large quantities. For information, contact Prentice-Hall, Inc., General Special Sales Division, Englewood Cliffs, N. J. 07632.

A SPECTRUM BOOK

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Editorial/production supervision by Louise M. Marcewicz and Marlys Lehmann
Cover design by Jeannette Jacobs
Manufacturing buyers: Cathie Lenard and Barbara A. Frick

ISBN 0-13-931220-X {PBK.}

ISBN 0-13-931238-2

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

This book is dedicated first to Marlene and Shandra. Without their support, none of the things I do would be possible. It is also dedicated to my supporters and mentors: My mother, Marian Derfler; Bill Mac Laren; Al Edmonds; Gil Sentimore; Jerry Landry; John and Judy Collins; and the others who "keep the faith." Nobody does it alone!

Contents

Acknowledgments

I would like to acknowledge the contributions of David Lindbergh and Bob Savage to this book. These two gentlemen are masters of things I barely understand.

Many terms used in this book are registered trademarks of various corporations or individuals. Here are the most common terms used:

CAT: Novation, Inc.
CP/M: Digital Research, Inc.
CROSSTALK: Microstuf, Inc.
Dumb Terminal: Lear Siegler, Inc.
Forum 80: Bill Abney
LYNX: Emtrol Systems, Inc.
Microconnection: Microperipheral Corp.
MNET: CompuServe
OMNITERM: Lindbergh Systems
SmartModem: Hayes Microcomputer Products
Source: Source Telecomputing Corporation
ST80, ST80-CC, ST80-PBB, ST80-X10: Lance Micklus, Inc.
TRS-80, Modem I, Modem II: Tandy Corporation
Z80: Zilog, Inc.

1

The Role of Data Communications

We are in the middle of humanity's second great information explosion. The first great explosion came with the invention of movable metal type. The second great explosion is being fueled by the marriage of computers and communication. Many experts believe the availability of inexpensive data communication systems will make as large an impact on our culture as did the development of the printing press in the 15th century. More things have been invented since World War II than during the total prior history of mankind (Figure 1-1). Information pours out of newspapers, books, and the electronic media; but the individual often finds that this information must be arranged in a usable form and transmitted in a timely manner to be useful. Sorting, arranging, and transmitting information are the strong features of data communication systems.

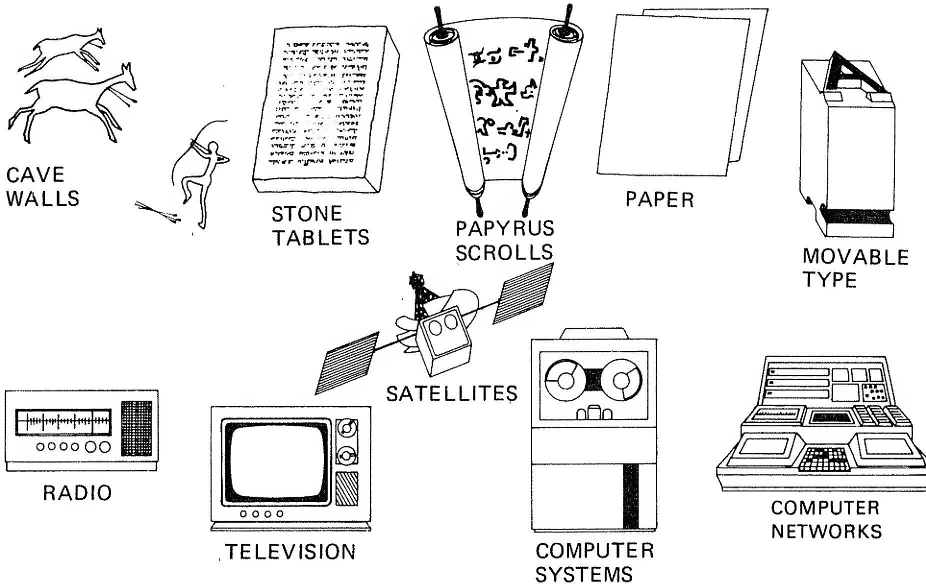


FIGURE 1-1. Development of data communication through the ages.

THE TRS-80: A COMMUNICATION DOORWAY

If you own a TRS-80, you have the prime element in a data communication terminal. It can allow you access to the world's information when and how you want it. On a practical level, giving your microcomputer a communication capability can make it easy to transfer programs from one system to another, even if the systems have different disk or cassette storage devices. On a broader scope, a microcomputer acting as a terminal can bring in the current news, transportation schedules, shopping and stock information, and mail at the speed of light.

The information network we used until the early 1980s had not changed much since the 1930s. It was based on the newspaper and supplemented by magazines, telephones, radio, and later, television. This sort of network ties its users into certain conditions. The major condition or prerequisite for receiving information out of the network is this: You must be connected to the network at the specific time the information is being sent. You

must get the right issue of the newspaper to read the story you want. You must be ready to listen to or watch the evening news at the time it is being transmitted or you will miss the item you want to see. You must be ready to answer the telephone within the 30-second period of the typical 7-ring attempted call or you will not be connected. Studies show the number of successfully completed telephone calls in the business world is regularly as low as a miserable 15%. The demand that you be plugged into the system at just the right time to receive a transmitted message is a kind of tyranny. I refer to it as the "Time Tyranny of Telecommunications" (Figure 1-2).

THE TYRANNY OF TIME

A great deal of research and money has been spent on products to break the Time Tyranny of Telecommunications. Telephone-answering machines and video tape recorders are devices that automatically connect into the network at the right time and save the transmitted messages until you are ready to play them back. They act as a kind of storage buffer between you and the distribution network. They buffer or hold the information until you are ready to receive it, but the network still transmits or broadcasts the information only when it is convenient or expedient for the individuals *providing* the information. The network does not know or care if the user is ready to listen.

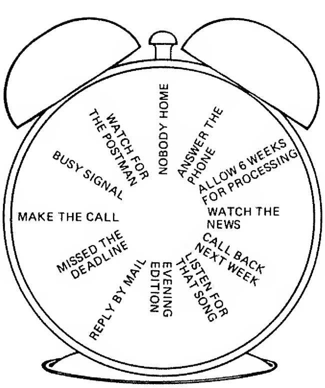


FIGURE 1-2. The Time Tyranny of Telecommunications.

A more modern approach is through modification of the network itself. The information network can be changed so it becomes an interactive system instead of just a broadcast into the blind. Information (messages, orders, news, sports, and so on) is loaded into an interactive system at the convenience of the information receiver and in the order and format suiting his needs.

These modern networks take many forms. Several telephone companies now provide centralized storage of voice messages. Voice messages are digitally recorded and played back when the person they are addressed to checks into his central "mailbox" from any telephone. Voice message-storage systems do not require any specialized equipment to send and receive, but the messages cannot be sorted, filed, or stored by the receiver in any easy way. The most effective modern information networks or systems can do all these things and more.

Three general kinds of systems are becoming popular: electronic message systems, information utilities, and teletext systems.

ELECTRONIC MESSAGE SYSTEMS

There are many kinds of electronic message systems. Among them are Computer Bulletin Board Systems (CBBS), Apple Bulletin Board Systems (ABBS), Forum 80 systems, PET Message Systems, and others. All these systems operate in a similar manner, but they differ in the hardware and software used by the central computer and in the features they may provide.

Physically, an electronic message system is usually a standard microcomputer from one of the popular manufacturers. The computer probably has 48 kilobytes (K) or more of random access memory (RAM) and several disk drives. Most importantly, it has a device called an auto-answer modem. A modem (which will be described more thoroughly in Chapter 4) is a device that converts the electrical output of a computer or terminal into audio tones which can be transmitted and received over standard telephone lines. This modem picks up or "answers" the telephone line when it rings. The caller on the distant end is automatically connected to the computer running the message system program.

Message-system programs differ in detail, but they usually perform several standard functions. They normally ask the caller first to complete a sign-on routine for purposes of identification. The sign-on routine may include many details such as name, location, and telephone number of the caller; or it may be as simple as a coded serial number transmitted automatically by the caller's microcomputer terminal.

After the sign-on, the message program gives the user several options. The first microcomputer-based message systems were little more than places where users could "pin up" notes to their friends. The most advanced systems now allow their users to run programs, play games, use higher order programming languages (such as Pascal), keep private mailboxes, read interesting articles or editorials, order products from other users, and exchange programs with the host system. These services are usually provided free of charge. Often they are operated by private individuals, but many businesses selling computer products have found a message system to be a valuable link to their customers.

INFORMATION UTILITIES

Would you like to have the complete reports of a national news wire service in your home? Would you like to have stock market reports, business analyses, government publications, home and garden articles, shopping tips, merchandise ordering, train and airline schedules, and mail service at your fingertips? Would you also like to have the option of using programs requiring large memories and programming languages not normally available on a microcomputer? Information utilities can provide you with all these capabilities and more.

Information utilities are a specialized form of what has been known for years as *time-sharing* services. In a time-sharing service, a large central computer (or often a cluster of medium-size computers) serves customers in remote locations over telephone lines. Each customer appears to have a dedicated system. The time-sharing services entering the information utility market provided some very unique "programs" for their customers to run. These

programs provide the kinds of "electronic Sunday newspaper" and reference library services described above.

Your TRS-80 can be used as an access device to these information utilities. A communicating microcomputer can enhance the operation of these subscription services by saving information and files locally, preparing messages and files for transmission to the bigger system, and speaking to the bigger system in quick shorthand codes.

Two major information utilities are marketing their services especially for microcomputer users. The Source is the name of one service provided by Source Telecomputing Corporation in McLean, Virginia. A competing service is marketed by CompuServe of Columbus, Ohio. Both of these services are described in detail in Chapter 10. While Source and CompuServe compete with each other, they both face growing competition from another form of information utility: teletext and videotext services.

TELETEXT AND VIDEOTEXT

Teletext and videotext systems also provide interactive delivery of information into the homes and offices of their subscribers. Since their services are not designed especially for microcomputer users, I will not spend much time describing them. However, anyone interested in data communications should understand basically how these systems work and what their impact might be.

Teletext and videotext are both centered on the same kind of centralized information-loaded computer used by the information utilities. These systems rely on terminal devices more common in the home than the microcomputer.

Teletext uses the home telephone line and a specially modified television set to link the subscriber to the central computer. The typical user has a small keypad which allows selections from displayed menus. The central computer is informed of the selection by a short message transmitted over the telephone line from the television/terminal. The selected page of information is sent from the computer to the home user over the telephone line at high speed.

Videotext also uses a modified television set, but not the telephone line. The pages of information are transmitted by a

local television station during idle milliseconds in the regular transmission signal. The information is sent in short but high-speed bursts repeated in a regular pattern. Videotext is not truly an interactive system. There are no transmissions from the television/terminal to the central computer. Instead, the user's television set waits for the desired page of information to be transmitted; then it captures and displays it. This wait is usually less than a minute.

Videotext and teletext systems are similar to information utilities, but they aim at slightly different markets. Videotext and teletext want to serve the average American family at home. Information utilities are probably aimed at more sophisticated users at home and in business. Both videotext and teletext systems make use of graphic color displays. They provide information quickly and in standard formats. They are simple to use, but they lack the flexibility and power gained from teaming centralized information utilities with remote microcomputer terminals.

TRANSMISSION SYSTEMS

The new modern interactive information networks are possible because of the availability of modern transmission systems. Transmission systems carry the electronic messages between communicating microcomputers and the terminals or computers to whom they are talking. The only pieces of the transmission systems we normally think of or see are the home telephone instruments and the wires going from pole to pole. These instruments and local wires are common to all the systems we use, but after our voice or computer modem tone leaves the local telephone office, it may go by many routes.

Frequently, users of electronic message systems find they make many long-distance telephone calls to get on the systems interesting to them. The cost of these calls can mount quickly. The most common long-distance routes are those provided by American Telephone and Telegraph. Over the past few years, however, some competition has developed due to the deregulation of the telecommunications industry in the United States. This means you may be able to make long-distance calls more cheaply by using an alternative to the traditional telephone service. Two

companies are competing strongly in providing services you can use. Southern Pacific Communications offers a service called "Sprint" to private customers. MCI Communications Corporation was the first company to challenge the American Telephone and Telegraph monopoly in long-distance communications; their service is known as Execunet. In Chapter 10, the geographic areas served by these carriers and their rate structures will be described. This is important information for the microcomputer communicator who wants to use electronic message systems around the country.

The information utilities also make use of special communication carriers to route the great volume of data they transmit and receive. Tymnet and Telenet are two such major telecommunications networks especially dedicated to computer communication. Carriers dedicated to a special purpose such as data communication are called value-added carriers because of the message processing, error detection and correction, and other specialized services they provide. These networks are not normally noticed by information utility users, but they function in very special ways to connect digital communication systems.

Finally, the spread of cable television services and the possibility of direct satellite transmission are creating new opportunities for videotext and modified teletext services. Several two-way cable television systems are operating around the country with exciting results. Presidential candidates, local elected officials, and professional athletic coaches have all experienced the instant feedback of two-way cable.

THE DIGITAL-DATA FUTURE

Data communication networks are rapidly changing the way we conduct our lives. These networks have made a tremendous impact on banking, publishing, travel, and government services. The microcomputer in your home or office can be an important part of one or several data networks. Opening the door to the warm winds of the information explosion is an easy and pleasant thing to do. The following chapters will give you the technical and practical keys to microcomputer communication.

2

The Fundamentals

It is not necessary to understand the internal combustion engine in order to drive a car; similarly it is not necessary to understand anything technical about data communication to use an electronic message system or information utility. Knowing what is going on "under the hood," however, can greatly increase your enjoyment and possibly reduce maintenance and repair bills. Some technical understanding is also needed when you are trying to customize your own machine.

SERIAL DATA

The data-transmission systems used by microcomputers are referred to as *serial systems.* This means that data comes out bit by bit in a serial stream, rather than the way some printers and

other devices are fed by the parallel transmission of eight bits of data simultaneously.

Every microcomputer has some input and output ports. The keyboard feeds an input port and the video monitor or RF modulator receives signals from an output port. The cassette or disk controller moves data both in and out and is therefore called an I/O port. It is possible to use the cassette port of most microcomputers to communicate over limited distances by connecting the audio signals to the telephone lines. This can only be done with two identical model computers, however, because cassette systems are not standard. But even when identical computers are used, the cassette-coding system is very sensitive to the changes imposed on the signal by the telephone circuit. Many computer users experience problems trying to load programs from a local recorder. Minor changes in audio level or the phase of the signal can make loading difficult. The problems are compounded when telephone lines are used. Use of the cassette port as a communication port can be a very limited and frustrating way to try to communicate.

RS-232-C

A standard and practical I/O coding scheme is needed for long-distance data communications. Computers communicate internally and externally in digital signals. Inside every system, direct current voltages are being switched from high voltage to low voltage many times a second. These changes in voltage represent digital bits of information. However, the voltages used differ between the systems. Even in the microcomputer family, different microprocessors use different voltage levels. If all systems are to communicate on a common network, some standard for external voltage levels must be set. We need a solid definition of the electrical standards to be used. This has been supplied by the Electronic Industries Association (EIA) standard code RS-232-C. Outside of the United States, a nearly identical code is known as the International Consultive Committee on Telephone and Telegraph (CCITT) code V.24. This code provides a common description of what the signal coming out of and going into the serial port will

look like electrically. Specifically, RS-232-C provides for a signal swinging from a nominal +12 to a nominal −12 volts at certain specified current levels and resistive loads (Table 2-1).

TABLE 2-1. RS-232-C Electrical Standards

Signal Name	*Logic State*	*Control Function*	*Electrical Parameters*
Mark	One	Off	Minus 24 to minus 3 volts
Space	Zero	On	Plus 3 to plus 24 volts

The standard also defines the cables and connectors used to connect data communication devices together. This will become quite important when we discuss the hookup of actual hardware. Using this standard code simplifies the job of getting information in and out of a computer, terminal, or peripheral device. A new standard, called RS-449, has been adopted which will eventually be a replacement or an alternative to RS-232-C, but compatibility with RS-232-C is specified in RS-449. RS-232-C will continue to be a useful signalling standard for many years to come.

The RS-232-C electrical standard limits the length of the connecting cable to 50 feet. In practice, cables many times that length can be used if they are routed away from appliances and other sources of electrical interference. The timing signals used in synchronous transmission are particularly susceptible to noise.

The voltage area between plus three and minus three volts is a transition area. Signals should not rest in this zone because they will then cause erratic operation of the logic circuits. The positive and negative signals do not have to be of the same amplitude. Positive voltages of 12 volts and negative voltages of −5 or −18 volts are commonly found working together. The change in voltage state is the actual signalling function.

The terms commonly used with electrical coding standards may be confusing because they are often carried over from other systems. Once you get them straight, however, they are easy to understand. Because of the peculiarities of solid state logic devices, a logic state called "0" may not indicate zero volts. Indeed, just the opposite is true. A logic state of 0 is defined as the positive voltage (+3 to +25 volts) signal in RS-232-C. This is also known

as a space signal. Space and mark are two designations held over from the days of mechanical printers, which are operated by electromagnets and driven by direct current circuits. If you read any literature stating that a space should be transmitted, you know you are looking for a positive voltage or a *0 logic state.*

The logic state of 1 is just the opposite. A *logical 1* is a negative direct-current voltage; it is also known as a *mark.* You may wish to remember the phrase, "The teacher gave the student a low mark, but a logical one."

The change in the direct-current voltage level serves as the signal or bit of information in the RS-232-C system. These bits are sensed, counted, and stored by data communication devices. The direct-current voltages used in this interconnection system can travel about 50 feet before they lose their important electrical characteristics, so RS-232-C signaling is not used directly for long-distance communication. (The actual maximum effective DC path depends on many factors such as speed, cable type, etc., but 50 feet is a standard for reliable high-speed service.) RS-232-C signalling is used to connect microcomputers with modems. RS-232-C is the most commonly used standard for local connection of microcomputers, terminals, and modems.

RS-232-C is, however, just an electrical standard. It defines the voltage swings and other electrical parameters. It does not define what the voltage swings mean in terms of intelligent information. It is as if we said we will all use red and blue flags for signalling, but we do not define what the position of the flags will mean. Another standard, a coding standard, is needed.

ASCII

The most commonly used coding standard is the American Standard Code for Information Exchange, or ASCII. ASCII is actually a data alphabet. Internationally, it is known as CCITT Alphabet Number 5. This alphabet not only tells us the coding of the electrical signals that make up the characters in ASCII, but also provides useful numeric values for the characters and special standards for recording information on punch cards and magnetic media such as tape. ASCII defines the standard keyboard and provides codes for

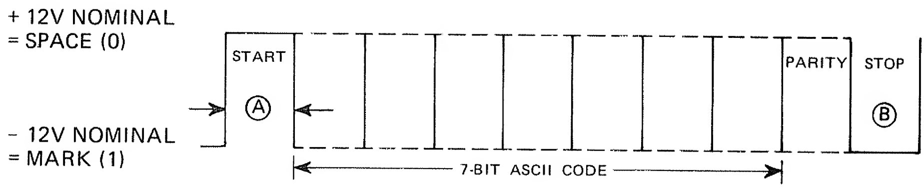
the smooth processing of information over data-transmission systems.

ASCII defines certain coded signals as *control codes.* These are codes that usually mean something special to a machine. Control codes may tell electronic printers to tab, ring a bell, or begin a new page. They may stop the running of a computer program or turn on a tape drive. Control codes are valuable in data-communication systems because of the flexibility they provide. The availability of control codes is of particular interest to microcomputer users because some microcomputer hardware and software combinations do not have the ability to transmit control codes when operating as terminals. Such systems would be severely limited when operating with certain message systems or running programs on computer time-sharing services.

The coding of an ASCII character is easy to understand. The code essentially signifies how many times and when a voltage or tone goes high or low in a 7-bit sequence. The capital letter A, for instance, has a coding of 1000001. This would be coded over a piece of wire by sending one negative voltage (remember, a logic state of 1 is a negative voltage in RS-232-C signalling), five positive voltage pulses, and a final negative voltage pulse within certain time slots set by internal clocks in the equipment. Over a telephone line, the direct-current voltages would be converted to audio tones by a modem. The coding of the audio tones for the letter A would be one low tone, five high tones, and one low tone. This coding is also a number in the binary (base 2) number system. If the binary value is converted to the decimal (base 10) number system, it becomes 65. This coding of a mark, five space signals, and a mark always represents A, and it always has a decimal value of 65 in the ASCII alphabet. The number value of an ASCII character or a string of ASCII characters is often used in programs or transmission systems for sorting or checking data (Figure 2-1).

OTHER ALPHABET CODES

You may sometimes hear of other codes used for the transmission of alphabetical characters. The simplest scheme, of course, is the International Morse code. In Morse, as it is commonly called,



Ⓐ BIT DURATIONS:
1200 BAUD = 833 MICROSECONDS
600 BAUD = 1667 MICROSECONDS
300 BAUD = 3.33 MILLISECONDS
150 BAUD = 6.67 MILLISECONDS
110 BAUD = 9.09 MILLISECONDS

Ⓑ 110 BAUD RATE FORMAT HAS 2 STOP BITS.
ALL OTHER RATES HAVE 1 STOP BIT IN FORMAT.

FIGURE 2-1. RS-232-C transmission of one character. This string of pulses represents one ASCII coded character transmitted using RS-232-C signalling. The width of the pulses will change directly with the baud rate. The actual voltages used may vary from +3 to +4 volts for a space to –3 to –24 volts for a mark.

characters are represented by long and short pulses of light, sound, or electricity called dots and dashes. Morse code still finds some use in high-frequency radio transmission, but now it is often sent and received by electronic terminals equipped with microprocessors.

Baudot code is a standard automatic printing machine code which represents each character by five bits of data instead of ASCII's eight. It is often referred to as a 5-level code. Machines using Baudot code are still common and Baudot networks serving deaf users are operating in many parts of the country.

The name "Baudot code" is commonly used to refer to the 5-bit coding scheme presented in Table 2-2. This particular code, however, was designed by Donald Murray and is quite different from Baudot's original 5-bit code. It is also known as the CCITT alphabet No. 2. Note that there are no real lower-case letters represented in the Baudot Code.

TABLE 2-2. Baudot Character Alphabet

CODE	*Lower Case*	*Upper Case*
Space 1 2 3 4 5		
1 1 0 0 0	A	–
1 0 0 1 1	B	?
0 1 1 1 0	C	:
1 0 0 1 0	D	$
1 0 0 0 0	E	3
1 0 1 1 0	F	,
0 1 0 1 1	G	&
0 0 1 0 1	H	British Pound
0 1 1 0 0	I	8
1 1 0 1 0	J	,
1 1 1 1 0	K	(
0 1 0 0 1	L	)
0 0 1 1 1	M	
0 0 1 1 0	N	.
0 0 0 1 1	O	9
0 1 1 0 1	P	0
1 1 1 0 1	Q	1
0 1 0 1 0	R	
1 0 1 0 0	S	Bell
0 0 0 0 1	T	5
1 1 1 0 0	U	7
0 1 1 1 1	V	;
1 1 0 0 1	W	2
1 0 1 1 1	X	/
1 0 1 0 1	Y	6
1 0 0 0 1	Z	"
1 1 1 1 1	LETTERS (Shift to Lower Case)	
1 1 0 1 1	FIGURES (Shift to Upper Case)	
0 0 1 0 0	SPACE	
0 0 0 1 0	CARRIAGE RETURN	
0 1 0 0 0	LINE FEED	
0 0 0 0 0	BLANK	

International Business Machines Corporation developed the "Extended Binary Coded Decimal Interchange Code" or EBCDIC. This 8-bit code is commonly used in connecting IBM equipment, particularly printing typewriters. Some software and hardware systems have been devised to allow communication between ASCII and EBCDIC speaking devices, but this is still not an easy task.

ASCII is the most commonly used coding system in microcomputer communication. Be wary of any equipment that does not speak this common language.

Table 2-3 presents ASCII and EBCDIC character alphabets. This table is not meant to be an exhaustive treatment of either alphabet, but rather to show some of the differences and similarities between these two common coding schemes. ASCII has more symbols and characters and can probably support non-English language symbols more easily. Both ASCII and EBCDIC contain unique codes used for transmission signalling, but they are not all the same.

TABLE 2-3. ASCII and EBCDIC Character Alphabets

ASCII Binary Code	*EBCDIC Binary Code*	*Character*
0000000	00000000	NUL
0000001	00000001	SOH
0000010	00000010	STX
0000011	00000011	ETX
0000100		EOT
0000101	00101101	ENQ
0000110		ACK
0000111		BEL
0001000		BS
0001001		HT
0001010		LF
0001011		VT
0001100	00001100	FF
0001101		CR
0001110		SO
0001111		SI

TABLE 2-3. ASCII and EBCDIC Character Alphabets **(continued)**

ASCII Binary Code	*EBCDIC Binary Code*	*Character*
0 0 1 0 0 0 0		DLE
0 0 1 0 0 0 1		DC1
0 0 1 0 0 1 0		DC2
0 0 1 0 0 1 1		DC3
0 0 1 0 1 0 0		DC4
0 0 1 0 1 0 1		NAK
0 0 1 0 1 1 0	0 0 1 1 0 0 1 0	SYN
0 0 1 0 1 1 1	0 0 1 0 0 1 1 0	ETB
0 0 1 1 0 0 0		CAN
0 0 1 1 0 0 1	0 0 0 1 1 0 0 1	EM
0 0 1 1 0 1 0	0 0 1 1 1 1 1 1	SUB
0 0 1 1 0 1 1	0 0 0 1 0 1 1 1	ESC
0 0 1 1 1 0 0		FS
0 0 1 1 1 0 1		GS
0 0 1 1 1 1 0		RS
0 0 1 1 1 1 1		US
0 1 0 0 0 0 0	0 1 0 0 0 0 0 0	SP
0 1 0 0 0 0 1	0 1 0 1 1 0 1 0	!
0 1 0 0 0 1 0	0 1 1 1 1 1 1 1	”
0 1 0 0 0 1 1		#
0 1 0 0 1 0 0	0 1 0 1 1 0 1 1	$
0 1 0 0 1 0 1	0 1 1 0 1 1 0 0	%
0 1 0 0 1 1 0	0 1 0 1 0 0 0 0	&
0 1 0 0 1 1 1	0 1 1 1 1 1 0 1	,
0 1 0 1 0 0 0	0 1 0 0 1 1 0 1	(
0 1 0 1 0 0 1	0 1 0 1 1 1 0 1	)
0 1 0 1 0 1 0		*
0 1 0 1 0 1 1	0 1 0 0 1 1 1 0	+
0 1 0 1 1 0 0	0 1 1 0 1 0 1 1	,
0 1 0 1 1 0 1	0 1 1 0 0 0 0 0	–
0 1 0 1 1 1 0	0 0 1 0 0 1 0 0	.
0 1 0 1 1 1 1	0 1 1 0 0 0 0 1	/
0 1 1 0 0 0 0	1 1 1 1 0 0 0 0	0
0 1 1 0 0 0 1	1 1 1 1 0 0 0 1	1

TABLE 2-3. ASCII and EBCDIC Character Alphabets **(continued)**

ASCII Binary Code	*EBCDIC Binary Code*	*Character*
0 1 1 0 0 1 0	1 1 1 1 0 0 1 0	2
0 1 1 1 1 0 0	1 1 1 1 0 0 1 1	3
0 1 1 0 1 0 0	1 1 1 1 0 1 0 0	4
0 1 1 0 1 0 1	1 1 1 1 0 1 0 1	5
0 1 1 0 1 1 0	1 1 1 1 0 1 1 0	6
0 1 1 0 1 1 1	1 1 1 1 0 1 1 1	7
0 1 1 1 0 0 0	1 1 1 1 1 0 0 0	8
0 1 1 1 0 0 1	1 1 1 1 1 0 0 1	9
0 1 1 1 0 1 0	0 1 1 1 1 0 1 0	:
0 1 1 1 0 1 1	0 1 0 1 1 1 1 0	;
0 1 1 1 1 0 0	0 1 0 0 1 1 0 0	< (LESS THAN)
0 1 1 1 1 0 1	0 1 1 1 1 0 1 1	=
0 1 1 1 1 1 0	0 1 1 0 1 1 1 0	> (MORE THAN)
0 1 1 1 1 1 1	0 1 1 0 1 1 1 1	?
1 0 0 0 0 0 0	0 1 1 1 1 1 0 0	@
1 0 0 0 0 0 1	1 1 0 0 0 0 0 1	A
1 0 0 0 0 1 0	1 1 0 0 0 0 1 0	B
1 0 0 0 0 1 1	1 1 0 0 0 0 1 1	C
1 0 0 0 1 0 0	1 1 0 0 0 1 0 0	D
1 0 0 0 1 0 1	1 1 0 0 0 1 0 1	E
1 0 0 0 1 1 0	1 1 0 0 0 1 1 0	F
1 0 0 0 1 1 1	1 1 0 0 0 1 1 1	G
1 0 0 1 0 0 0	1 1 0 0 1 0 0 0	H
1 0 0 1 0 0 1	1 1 0 0 1 0 0 1	I
1 0 0 1 0 1 0	1 1 0 1 0 0 0 1	J
1 0 0 1 0 1 1	1 1 0 1 0 0 1 0	K
1 0 0 1 1 0 0	1 1 0 1 0 0 1 1	L
1 0 0 1 1 0 1	1 1 0 1 0 1 0 0	M
1 0 0 1 1 1 0	1 1 0 1 0 1 0 1	N
1 0 0 1 1 1 1	1 1 0 1 0 1 1 0	O
1 0 1 0 0 0 0	1 1 0 1 0 1 1 1	P
1 0 1 0 0 0 1	1 1 0 1 1 0 0 0	Q
1 0 1 0 0 1 0	1 1 0 1 1 0 0 1	R
1 1 0 0 1 0 1	1 1 1 0 0 0 1 0	S

TABLE 2-3. ASCII and EBCDIC Character Alphabets **(continued)**

ASCII Binary Code	*EBCDIC Binary Code*	*Character*
1010100	11100011	T
1010101	11100100	U
1010110	11100101	V
1010111	11100110	W
1011000	11100111	X
1011001	11011000	Y
1011010	11011001	Z
1011011		[(LEFT BRACKET)
1011100		\ (LEFT SLASH)
1011101		] (RIGHT BRACKET)
1011110		↑ (CARET OR UP ARROW)
1011111	01101101	———
1100000	01111101	,
1100001	10000001	a
1100010	10000010	b
1100011	10000011	c
1100100	10000100	d
1100101	10000101	e
1100110	10000110	f
1100111	10000111	g
1101000	10001000	h
1101001	10001001	i
1101010	10010001	j
1101011	10010010	k
1101100	10010011	l
1101101	10010100	m
1101110	10010101	n
1101111	10010110	o
1110000	10010111	p
1110001	10011000	q
1110011	10011001	r
1110011	10100010	s
1110100	10100011	t
1110101	10100100	u

TABLE 2-3. ASCII and EBCDIC Character Alphabets **(continued)**

ASCII Binary Code	*EBCDIC Binary Code*	*Character*
1 1 1 0 1 1 0	1 0 1 0 0 1 0 1	v
1 1 1 0 1 1 1	1 0 1 0 0 1 1 0	w
1 1 1 1 0 0 0	1 0 1 0 0 1 1 1	x
1 1 1 1 0 0 1	1 0 1 0 1 0 0 0	y
1 1 1 1 0 1 0	1 0 1 0 1 0 0 1	z
1 1 1 1 0 1 1		
1 1 1 1 1 0 0	0 1 1 0 1 0 1 0	
1 1 1 1 1 0 1		
1 1 1 1 1 1 0		
1 1 1 1 1 1 1		DEL

SOH: Start of Heading
STX: Start of Text
ETX: End of Text
EOT: End of Transmission
ENQ: Enquiry
ACK: Acknowledge
NAK: Negative Acknowledge
NUL: Null (non-printing fill)
HT: Horizontal Tab
VT: Vertical Tab
LF: Line Feed
FF: Form Feed
CR: Carriage Return*
BS: Back Space
DC1, DC2, DC3, DC4: Unspecified Controls usually used "as needed" by manufacturers.

*Note that a carriage return moves the cursor or typing head back to the first position in a line. It does not move it down one line as is automatically done on a typewriter. A carriage return is usually accompanied by a line feed to move the activity to the next line. Nulls are often inserted to allow print heads the physical time needed to perform a carriage return and line feed before the next character is sent.

SYNCHRONOUS AND ASYNCHRONOUS TRANSMISSION

During the transmission of ASCII characters, we need some way to tell when a character starts and stops. Two formats are used, one called *asynchronous* and the other called *synchronous.* Both formats require some additional information called *framing bits* to be sent in order to tell when a character is starting or stopping.

The asynchronous format requires such framing information to be sent with each character. Synchronous transmission gathers up blocks of characters and only defines the beginning and end of each block. Synchronous transmission is slightly faster and more efficient and is better suited to high-speed transmission systems handling thousands of characters per second. Asynchronous transmission is the standard format used with microcomputer-based systems.

The asynchronous format adds a *start bit* to each character to be transmitted. The start bit is a logical zero, or space, and is represented by the positive voltage level on the RS-232-C direct-current line. The start bit tells the receiving system to look at the next bits as an ASCII character. At the end of the ASCII character, an eighth bit is added for what is termed the *parity check.* This parity bit is a form of error detection and correction, but it is seldom used in microcomputer communications systems. In sophisticated systems, the parity bit is given a value (1 or 0) that will make the sum of the ones in the 8-bit word come out to meet a predetermined standard—either odd or even. This serves as a constant check on the quality of the transmission. If the parity check is not the same as the transmitter and receiver have been programmed to expect, various corrective actions can be taken. After the 7 bits of ASCII character and the parity bit are sent, one or more *stop bits* (logic 1) are transmitted. These stop bits insure that the receiver recognizes the next start bit and the whole process starts over again, many times a second. The rate of transmission determines how many stop bits are transmitted. Two stop bits are usually used at lower speeds and one during faster transmission. Transmission speeds for microcomputer systems are usually designated as either 110 or 300 baud.

BITS AND BAUDS

Transmission speeds can be described in four ways. The particular description depends on what kind of work you are doing with a data communication system. The most common term used on electronic message systems and information utilities is baud. The baud rate is simply the measure of transmission speed. Baud rate is not the same as bits per second, and the two cannot be directly

substituted for each other. It is like describing a stream of water. You can talk about the water moving at so many feet per second or you can talk about so many gallons passing a certain point in a second. They are two very different measures of the flowing stream. Baud rate is the equivalent of velocity. Mathematically, it is the reciprocal of the time duration of the shortest signal element in a transmission. This signal element in RS-232-C ASCII signalling is the time width of one RS-232-C bit. That time width for 300 baud works out to a little over 3.3 milliseconds. That means that each bit or voltage pulse is a little over 3.3 milliseconds in length. That's quick, but it is nothing compared to the state of the art in commercial circuits. Many circuits regularly operate at 1200, 2400, and 9600 baud.

Bits per second is a measure of information transfer. It includes only information elements, not start/stop bits. The information element in our systems would be the 7-bit ASCII characters. This is a more practical measure of the information actually getting through a system. Bits per second is directly related to characters per second. Simple math tells us we need only divide bits per second by 7 (bits in an ASCII character) to find characters per second. Another measure of information transfer is words per minute. Usually, a word consists of six letters (some formulas say five). Conversion from bits per second (PBS) to words per minute (WPM) would be done by:

$$\frac{\text{BPS}}{(7 \times 6)} \times 60 = \text{WPM}$$

This WPM figures is useful for persons trying to interface to printing terminals.

The transfer of real information, separated from all of the framing and control information, is also referred to as *throughput*. Throughput can be measured in either bits per second, characters per second, or words per minute.

Next: The Serial Port

This chapter has provided you with some of the fundamentals of data signalling and transmission. The next chapter will describe the process of creating the serial stream in terms of both hardware and software.

3

The TRS-80 Serial Port

This chapter will describe the serial port of the TRS-80 microcomputer systems. The initial discussion will look at the serial port from a functional hardware viewpoint. The later sections will discuss how the serial port relates to the Z-80 CPU in the model I and III computers. The discussion is also relevant for the Model II and Color Computer systems.

PARALLEL/SERIAL CONVERSION

A serial data communication channel is rather like a two-lane tunnel through a mountain which links two sections of a sixteen-lane highway. At each end, the traffic must funnel from eight lanes into one lane as it enters and fan out again as it emerges at

the other end. This process will limit the speed of the traffic in the tunnel since as many as eight lanes of traffic must narrow down to one. The alternative is a sixteen-lane tunnel, which would, of course, be prohibitively expensive. Similarly, sixteen parallel wires (lanes) would be costly between two computers (or other devices) over long distances, so a two-wire serial method was devised to make cheap long-distance data communication possible.

Before looking into the electrical device that makes this possible, we must stretch the tunnel example into a rather unreal scenario where all cars move in rows on the eight-lane highway in unevenly spaced clusters. As each cluster approaches the tunnel entrance, it must line up in single file. The car in the far right lane leads the way, with each successive car to the left queuing up behind the car which had been to its right. In the tunnel all cars travel at the same speed, and the spacing between cars in every cluster is the same. As the cars emerge from the tunnel, the first in line moves over to the far right lane; succeeding cars line up to the left of their predecessors and move on together down the eight-lane highway. Naturally, such a rigidly structured traffic pattern would be impossible to choreograph with cars, but it is essential for successful serial data transmission.

The electrical device that controls each "tunnel entrance" is actually a very capable "traffic cop." This device is often called a Universal Asynchronous Receiver/Transmitter–UART, and is pronounced YOU-art for short. It is also called an Asynchronous Communications Interface Adapter, or ACIA. More recent versions are the Universal Synchronous/Asynchronous Receiver/Transmitters (USARTs), the Protocol Controllers, and an extraordinary development known simply as a Serial Input/Output (SIO) device. Regardless of what these devices are called, they all perform serial-to-parallel and parallel-to-serial conversions with minimal supervision by the host computer.

Because of the so-called intelligence of one of these devices, very little supervision is required. The very latest models introduced to support the new 16-bit computers are actually highly specialized programmable microprocessors in their own right.

Like our surrealistic highway and tunnel, a UART has two separate directions of traffic. The path that converts a parallel data output to a serial bitstream is called the transmit channel. The path that converts an incoming serial bitstream back into a

parallel data input is called the receive channel. A very elaborate set of sensors and controls keep the data traffic in precise, syncopated rhythm.

THE WESTERN DIGITAL TR1602 UART

To illustrate how a UART works, let us look at a real device: the Western Digital TR1602. From the outside it looks just like hundreds of other chips mounted in insect-like packages called DIPs (short for Dual Inline Package—they have two parallel in-line rows of pins). The TR1602 has 40 pins, 20 in each row. As we describe what each pin does, remember that it is not very important to memorize the pin number for each function; you can always look up this information on the specification sheets if necessary. The pin designation given here relate to the UART itself. When the device is used in the TRS-80, different logical and circuit designations are given to the pins. This can create a little confusion if you don't understand that the UART control lines have been called one thing by the device manufacturer and another by the TRS-80 circuit designer. There are three separate types of signals which are applied to the UART's pins: data, status, and control. Data signals in both serial and parallel form are manipulated within a UART and transformed from parallel to serial and back. Status signals indicate the progress of these transformations and signal any error conditions which occur. Control signals provide both internal and external sequence coordination. Since device power and ground voltages are supposed to be constant, they are not considered signals per se.

The TR1602 requires two power supply voltages: +5V and −12V. There is a pin for each of these voltages, plus a third one which is tied to ground. One of these (the +5V supply) is the main power pin. The other is used to provide a bias voltage, which is used inside the device as a sort of reference. Very little power is required for the bias supply, but it must be present. This extra power supply costs money, so most new successor devices to the TR1602 are designed to operate with just one power supply.

The transmitter side of the TR1602 has 8 parallel transmit inputs (TR0-TR7) and 1 transmit data output (TRO). The con-

version is done using a shift register, which is a formation of memory circuits that are loaded all at once (parallel) and emptied one at a time (serial). As each bit is transmitted out, those behind it shift into the memory circuits in front of themselves. The ripple effect gives the appearance of motion out of the device.

Because many internal operations are required to convert a parallel 8-bit input to a serial output, a transmit clock (TRC) pin is provided. This is not the computer system clock. There is no synchronization between the UART and the host, hence the term *asynchronous.* The frequency of the TRC signal sets the baud rate of the transmitted serial bitstream. The baud rate is the rate at which the serial 1s and 0s come out of the device. Many standard rates are used. For most serial transmissions over common telephone lines using standard Bell 103 compatible equipment, 300 baud is used. The TRC signal is almost always 16 times the baud rate; for 300 baud the frequency of TRC is 4800 Hz (cycles per second). (See Figure 3-1.)

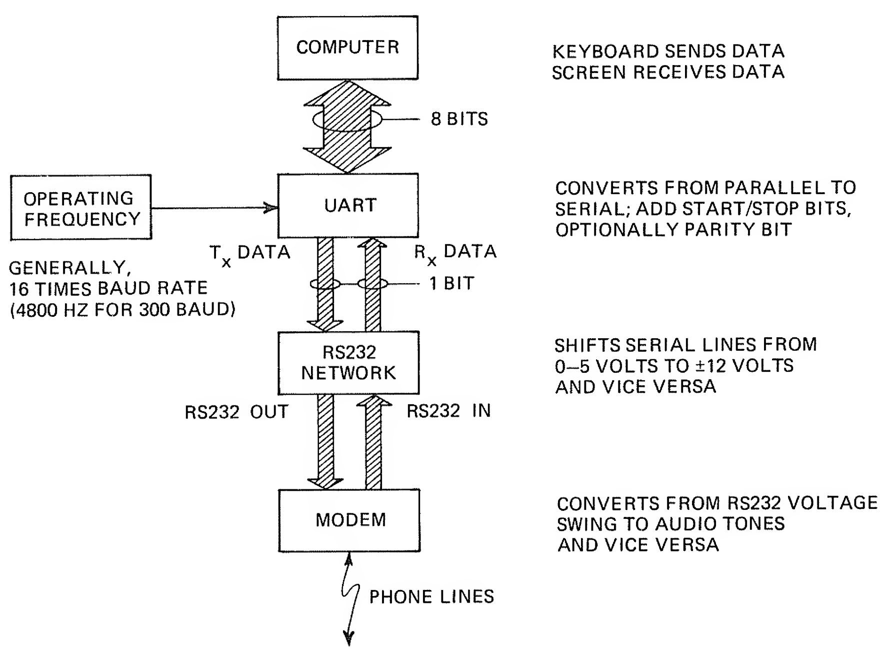


FIGURE 3-1. Telecommunication flow.

There is an inverse set of operations on the receive part of the chip, and the corresponding signals are *receive data input* (RI), *parallel data outputs* (RR0-RR7), and a *receive clock* (RRC) control signal. Frequently, TRC and RRC come from the same oscillator, but this is not mandatory. The two directions can operate independently at different baud rates, much as our tunnel could have a different speed limit in each direction.

There are two internal 8-bit parallel buffers that isolate the transmit and receive data holding registers from the computer system's data lines. Two data strobe signals are used to initiate data transfers into and out of the holding registers at the appropriate times. These control signals, *transmit buffer load* (THRL*) and *parallel data read* (RRD), are supplied by the host computer's serial device "driver" program. When parallel data is output and latched on the system's data lines, a strobe signal will immediately follow. This is applied to the TR1602's THRL* line, opening up the internal lines which connect the transmit buffer (which has latched onto the data) and the transmit register. The parallel-to-serial conversion circuits use the transmit register as their data source. On the receive side, the RRD signal will open up the connecting lines between the receive register and the parallel data receive buffer. This must be timed properly to avoid collisions on the system's data bus. There are three status signals which govern when these data strobes should be fired. When the transmitter section's parallel-to-serial process is complete, the *transmit register empty* (TRE) line goes to a logic 1. This is the signal that the host computer will look for before it sends more data and the THRL* strobe. When the data moves from the transmit buffer to the transmit register, the buffer is cleared, which sets the *transmit buffer empty* (THRE) line to a logic 1. THRE is used by the serial device routine to signal the arrival of data sent to the TR1602 upon receipt of the TRE signal and can be used to trigger the THRL* strobe.

Similarly, when the receiver section's serial-to-parallel process is complete, the TR1602 will output a logical 1 on the data received (DR) line. When ready to read this data, the host computer will output the RRD signal and read the data only when DR is 1. If parity is active, however, there is a safety status signal called *data received reset* (DRR*), which is set to a logic 0 if the parity-sensing circuits found an error and cleared (set to zero) the receive register. The presence of this DRR* status signal will abort

the data read operation if it cancels the effect of the DR status signal.

There are two control registers (one transmit, one receive), which are used to define the number of data and stop bit and the use and sense (odd/even) of the parity bit. Each of these programmable features is set by control signals on the *word length select* (WLS1 & WLS2), *parity inhibit* (PI), *even parity enable* (EPE), and *stop bit select* (SBS) pins. The two-word-length select pins are used as a binary pair to set the word length to 5 (0,0), 6 (0,1), 7 (1,0) or 8 (1,1) bits. The parity function will be inhibited if a logic 1 is applied to the PI line. If parity is used, a logic 1 on the EPE line will generate even parity; a logic 0 will result in odd parity. Two stop bits will be generated with a logic 1 on the SBS line, unless a 5-bit word length is selected, in which case a logic 1 will generate the standard 1.5 stop bits used by the 5-bit Baudot code. A logic 0 on the SBS line will generate just one stop bit.

To prevent inadvertent redefinition of any of the control functions, the TR1602 will not accept any changes at its WLS1, WLS2, SBS, PI, or EPE pins unless the *control load* (CRL) pin is momentarily strobed to a logic 1.

THE SERIAL STREAM

Each cluster in the serial bitstream is distinguished from its neighbors by start and stop bits which frame it. Usually one start bit signals the beginning of a cluster and two stop bits identify its end. Sometimes a parity bit is included after the data bits and before the stop bits, when transmission errors must be detected before causing damage to later data.

There are usually 8 data bits in a cluster. Fewer may be used if both the sender and the receiver permit it. In this form of transmission, the first and most basic requirement is that both ends expect the same number of data bits. The decision to use or to delete the parity bit follows in importance. Either 1 or 2 stop bits can be selected. One stop bit is usually used at 300 baud and higher, 2 stop bits at 110 baud.

Three status signals are generated upon the receipt of every cluster to guarantee that all three of the transmission parameters

are properly set and that no undetected errors occur. These are the *overrun error* (OE), the *framing error* (FE), and the *parity error* (PE). The first two signals are set to logic 1 levels only if an incorrect number of data bits are received, or an incorrect number of stop bits are received. Errors of this type after communications have been established are rare. The last status signal, the parity error, is set to a logic 1 only if the parity of the received data bit string (odd or even) does not match the parity bit. Parity, if used, can be preselected either even or odd.

If even parity is selected, the parity bit will be set to a logic 1 only when there are an even number of logic 1s in the data cluster. Conversely, odd parity would set the parity bit to 0 under the same conditions. Parity error signals are commonly used to trigger data retransmission. Because the parity bit (when selected) is found in every cluster, its use is often wasteful overhead.

If the parity bit is not used, the maximum length of each cluster is reduced from 12 to 11 bits—a 9% increase in the throughput of the link. This is such a significant benefit that parity is reserved for the very few applications where absolute accuracy is required (medical, nuclear control, and cryptological equipment, for example).

Continuous reading of the three status signals is not necessary. They are normally read at the beginning of each serial-to-parallel conversion cycle. A *status flag disconnect* (SFD) pin, held at a logic 1, can be used to disable OE, FE, and PL signals. Then, by pulsing the SFD signals to a momentary logic 0, current samples of these three signals can be "strobed" to the appropriate sensing circuits. In this way, overall power consumption of the TR1602 can be reduced by cutting off current flow to pins when they are not being used.

Like most complex programmable digital devices, the TR1602 has a master reset (MR) pin. Its function is to reset (clear) both buffer registers. With the buffers empty, the TRE, THRE, and TRO signals will go to logic 1 levels automatically. It also resets the three error-status indicators OE, FE, and PL plus the safety signal DRR*. This signal is generated by the software of the host computer during device initialization.

The TR1602 description given here is provided as an example of how the device typically functions in a data-communications system. The following description gives more detail on how

it relates to the Z-80 CPU in the TRS-80s. The pin designations are changed to match the specific circuit, but the functions are as described previously.

PROGRAMMING THE TRS-80 UART

The TRS-80 Model I computer has an optional RS-232-C interface board that connects to the Expansion Interface through a 42-pin connector inside a special compartment reserved for this board. The board has eight DIP switches on it, which together are called the sense switches and are normally used for entering the settings of baud rate, parity, number of data bits, and number of stop bits that the user desires. You may choose to use these, or you can ignore them and configure that UART with your own software.

Also on the board this is a large two-position switch marked "TERM" and "COMM." This switch is for swapping pins 2 and 3 of the RS232 connector, Transmit Data and Receive Data, so that your TRS-80 may be configured as a terminal device, which transmits on pin 2 and receives on pin 3, or as a communication device, reversing the functions of these pins. Most TRS-80 users connecting to a modem will leave this switch in the "TERM" position. The only time you might want to place the switch in the "COMM" position would be when your TRS-80 is connected to another microcomputer or is connected to a terminal or a printer.

Initializing the UART

The UART device on the RS232 board maps into four I/O ports on the TRS-80's Z-80. They are:

Port address (Hex)	*OUT*	*IN*
E8	Master Reset	Modem Status Register
E9	Baud Rate Select	Sense Switch
EA	UART Control	UART Status
EB	Transmit Data	Receive Data

Before the UART can be configured, you must send it a master reset, so that it will be in a known state that is proper for configu-

ration. This is done by writing any byte (the value does not matter) to port E8, the master reset port. In Z-80 assembler, this is:

```
OUT     (0E8H),A          ;MASTER RESET TO UART
```

Once you've done this, the UART can be set up. This is done by sending data to port E9 for baud-rate selection and sending data to port EA to set the UART for parity, number of stop and data bits, setting break, RTS, and DTR.

Setting Baud Rate

Baud rate is set by sending the proper nibble to port E9, the baud-rate select port. Each possible nibble, from 0 to F, selects a different baud rate. The high-order nibble determines the transmit baud rate and the low-order nibble determines the receive rate. If you wish, you can set up the UART for different baud rates on receive and transmit, but for most applications, the rates will be the same. The nibbles determine the baud rate as follows:

Nibble	*Baud Rate*	*Nibble*	*Baud Rate*
0	50	8	1800
1	75	9	2000
2	110	A	2400
3	134.5	B	3600
4	150	C	4800
5	300	D	7200
6	600	E	9600
7	1200	F	19200

You must create a byte with the proper nibbles for your transmit and receive rates and send it to port E9. For example, the byte 55 would set both receive and transmit rates to 300 baud. Or, 27 would set the transmit rate to 110 baud and the receive rate to 1200 baud. An example:

```
LD      A,77H           ; 1200 BAUD IN AND OUT
OUT     (0E9H),A        ; WRITE TO BAUD RATE SELECT
```

Now the UART has been reset and the baud rate selected. The next step is to set the UART's parity, number of data and stop bits, RTS, and DTR.

Configuring the UART Control Register

Port EA is the UART control register and the bit map is:

```
OUT     (0E8H),A        ; MASTER RESET TO UART
```

Once you've done this, the UART can be set up. This is done by sending data to port E9 for baud rate selection and sending data to port EA to set the UART for parity, number of stop and data bits, setting break, RTS, and DTR.

Setting Baud Rate

Baud rate is set by sending the proper nibble to port E9, the baud-rate select port. Each possible nibble, from 0 to F, selects a different baud rate. The high-order nibble determines the transmit baud rate and the low-order nibble determines the receive rate. If you wish, you can set up the UART for different baud rates on receive and transmit, but for most applications, the rates will be the same. The nibbles determine the baud rate as follows:

Nibble	*Baud Rate*	*Nibble*	*Baud Rate*
0	50	8	1800
1	75	9	2000
2	110	A	2400
3	134.5	B	3600
4	150	C	4800
5	300	D	7200
6	600	E	9600
7	1200	F	19200

You must create a byte with the proper nibbles for your transmit and receive rates and send it to port E9. For example, the byte 55 would set both receive and transmit rates to 300 baud. Or, 27 would set the transmit rate to 110 baud and the receive rate to 1200 baud. An example:

```
LD      A,77H         ; 1200 BAUD IN AND OUT
OUT     (0E9H),A      ; WRITE TO BAUD RATE SELECT
```

Now the UART has been reset and the baud rate selected. The next step is to set the UART's parity, number of data and stop bits, RTS, and DTR.

Configuring the UART Control Register

Port EA is the UART control register and the bit map is:

Data Bit	*Name*	*Function*
D0	Data Terminal Ready	1=Reset DTR, 0=Set DTR
D1	Request To Send	1=Reset RTS, 0=Set RTS
D2	Break	1=Normal, 0=True Break Send
D3	Parity Enable	1=Parity Disable, 0=Enable
D4	Stop Bit Select	1=2 Bits, 0=1 Bit
D5	Word Length 2	1=Add 2 to Word Length
D6	Word Length 1	1=Add 1 to Word Length
D7	Even Parity	1=Even Parity, 0=Odd Parity

Bits 0 and 1 of the UART control byte determine the status of the DTR and RTS output lines. If the bits are 1, the lines will be reset, and if 0, the lines will be set. Bit 2 controls the true break function. If 1, the UART will transmit normally, but if 0, it will send a break. To send a break, you should save the byte that you originally send to the UART control register. When you want to send a break, take this byte and reset bit 2. Send this new byte to the UART control register, wait about one-half second, and then put

back the original byte to stop the break and resume normal operation.

Bit 3 of the control byte allows you to either enable or inhibit parity. If bit 3 is set, parity will be disabled and there will be no parity bit regardless of bit 7 which controls even/odd parity. If bit 3 is reset, then there will be a parity bit, and its mode will be determined by bit 7.

Bit 4 controls the number of stop bits placed at the end of each byte. If 1, two stop bits will be sent, and if 0, one stop bit will be appended.

Bits 5 and 6 together control the number of data bits in each word. The minimum number of data bits that can be sent is five, so bits 5 and 6 determine how many additional bits to the five will be sent. If bit 6 is set, one extra bit will be sent, and if bit 5 is set, two extra bits will be sent. If both are set, then the sum of three extra bits onto the original five, a total of eight data bits, will be sent. Finally, if neither bit 5 nor 6 is set, then no extra bits will be sent, and only the original five will be used. Here is a table of the possibilities:

# of Data Bits	*Bit 5 – Add 2*	*Bit 6 – Add 1*
5	0	0
6	0	1
7	1	0
8	1	1

Bit 7 of the UART control word is used to determine even or odd parity. It is only used if bit 3 has been reset to enable parity. If bit 7 is 1, even parity will be used, and if it is 0, odd parity will be used. Once the data word with the correct bits has been created, it should be written out to port EA. For example:

```
LD      A,10110011B       ; EVEN, 7 BITS, 1 STOP
OUT     (0EAH),A          ; OUT TO UART CONTROL REG.
```

The UART is now configured properly and is ready for use. Often you will want to configure the UART from the DIP sense switches

rather than strictly with software. This means you must read the sense switches at port E9 and use the data to create a byte for the UART control register and for the baud rate select register.

Configuring the UART from the Sense Switches

The sense switches have no real function themselves, except that they can be read into port E9 and used by software to determine the configuration the UART should be set to. Each of the eight two-position switches on the DIP are labeled "OPEN" or "CLOSED." When a switch is in the "OPEN" position, it will be read by port E9 as a logic 1. When a switch is closed, it will read 0. The switches are not read in the same order as they are on the DIP, but instead in an predetermined pattern defined by Radio Shack. It is:

Switch #	*Data Bit*	*Data Bit*	*Switch #*
S1	D7	D0	S8
S2	D5	D1	S7
S3	D6	D2	S6
S4	D4	D3	S5
S5	D3	D4	S4
S6	D2	D5	S2
S7	D1	D6	S3
S8	D0	D7	S1

The meanings of the switch settings are also defined by Radio Shack. They are by function (C=Closed, O=Open):

Baud Rate	*S6=D2*	*S7=D1*	*S8=0*
110	C	C	C
150	C	C	C
300	O	C	C
600	O	C	O
1200	C	O	C
2400	C	O	O

Baud Rate	*S6=D2*	*S7=D1*	*S8=0*
4800	O	O	C
9600	O	O	O

Parity Enable	*S4=D4*
Parity Enable	C
Parity Disable	O

Stop Bits	*S5=D3*
One Stop	C
Two Stop	O

Word Length	*S2=D5*	*S3=D6*
5 Bit Word	C	C
6 Bit Word	C	O
7 Bit Word	O	C
8 Bit Word	O	O

Parity Select	*S1=D7*
Odd Parity	C
Even Parity	O

When an actual byte is read in from port E9, the data bits look like:

Data Bit	*Switch*	*Name*	*Function*
D0	S8	Baud Rate 2	Baud Rate Select Bit
D1	S7	Baud Rate 1	"
D2	S6	Baud Rate 3	"
D3	S5	Parity Enable	1=Parity Disable, 0=Enable
D4	S4	Stop Bit Select	1=2 Bits, 0=1 Bit
D5	S2	Word Length 2	1=Add 2 to Word Length
D6	S3	Word Length 1	1=Add 1 to Word Length
D7	S1	Even Parity	1=Even Parity, 0=Odd Parity

Before reading the sense switches, be sure you send a Master Reset to the UART as described above.

Now you can set the UART's baud rate from the sense switches. Read them in from port E9 and mask out all but the three baud rate bits D0-D2. This is most easily done by ANDing the byte with 7. For example:

```
OUT     (0E8H),A        ; DO MASTER RESET
IN      A, (0E9H)       ; READ SENSE SWITCHES
AND     7               ; MASK OUT ALL BUT BAUD BITS
```

After you have done this, you should have a value in the A register that corresponds to the three sense switches that are used to determine baud rate. Here is a table of the value you should have against the baud rate selected by the switches:

Value of switches	*Baud Rate*	*Nibble*
0	110	2
1	1200	7
2	300	5
3	4800	C
4	150	4
5	2400	A
6	600	6
7	9600	E

To get the correct nibble to be sent to the UART for the value of switches you have, you can use a table of nibbles in memory. For example, supposing the value of the switches is in the A register (as in the previous example), this routine will place the correct byte (both nibbles identical) in the baud-rate select register:

```
LD      E,A             ; LOW BYTE OF DE HAS VALUE
LD      D,0             ; HIGH BYTE OF DE IS 0
LD      HL,TABLE        ; GET TABLE ADDR
```

```
         ADD     HL,DE        ; HL HAS ADDR OF CORRECT BYTE
         LD      A,(HL)       ; GET BYTE IN A
         OUT     (0E9H),A     ; WRITE TO BAUD RATE SELECT

TABLE:   DEFB    22H          ; 110 BAUD
         DEFB    77H          ; 1200 BAUD
         DEFB    55H          ; 300 BAUD
         DEFB    0CCH         ; 4800 BAUD
         DEFB    44H          ; 150 BAUD
         DEFB    0AAH         ; 2400 BAUD
         DEFB    66H          ; 600 BAUD
         DEFB    0EEH         ; 9600 BAUD
```

As you can see, the nibble was found by crossreferencing a table of bytes in memory. This is a common method for programming the UART baud rate from the sense switches. Now you will want to write the UART control register from the sense switches. You will notice the similarity between the tables given for the UART control register and that for the sense switches. They are identical in bits D3-D7, so all you must do is read the sense switches from port E9 and mask out the baud rate bits D0-D2 by ANDing with FB. You must now set the lower 3 bits to the configuration you desire for break, RTS and DTR, as described above. The byte is then written out the UART control register port EA. For example:

```
IN      A,(0E9H)     ; READ SWITCHES
AND     0F8H         ; MASK OUT BAUD RATE DATA
OR      111B         ; SET LOWER 3 BITS
OUT     (0EAH),A     ; WRITE OUT TO UART CONTROL
```

You have now figured the UART to the switch settings and can use it for transferring data.

Using the Configured UART

To use the configured UART, you only need to deal with three ports, the Modem Status Register, port E8, which has status infor-

mation from the RS-232 device; the UART Status Register, which has status information on the UART itself; and port EB, the port where transmitted and received data bytes go.

The Modem Status Register at port E8 returns the CTS signal in D7, the DSR line in D6, Carrier Detect from pin 8 of the DB-25 connector in D5, and Ring Indicator from pin 22 of the DR-25 in D4. This port is read when you want to monitor the status of the special control lines on the RS-232 interface of some device that considers these lines important. Many common RS-232 devices such as modems do not use these lines and in applications where these are unimportant you may never need to read this port. The bit map of this port is:

Data Bit	*Use on Modem Status Register*
4	Ring Indicator–Pin 22
5	Carrier Detect–Pin 8
6	Data Set Ready–Pin 6
7	Clear to Send–Pin 5

Transmitting Data Bytes

Before writing any byte to the data register to be transmitted, you should read the UART status register at port EA and check to see if the previous byte has been fully transmitted. This is done by looking at bit 6 of the status register. If bit 6 is 1, then the transmitter holding register is empty, and you can put in a new byte to be sent. If bit 6 is 0, you must wait until it goes high before writing the byte. As soon as bit 6 goes high, you can write out the byte to port EB to be transmitted. For example, to send out the byte in A:

```
        LD    C,A          ; SAVE A IN C
WAIT:   IN    A,(0EAH)     ; READ UART STATUS
        AND   01000000B    ; LOOK AT BIT 6
        JR    Z,WAIT       ; REGISTER NOT EMPTY! WAIT.
        LD    A,C          ; GET A BACK FROM C
        OUT   (0EBH),A     ; WRITE OUT DATA
```

Receiving Data Bytes

To check if a data byte has been received, you can read the UART status port and check bit 7. If it is 1, a data byte has been received but not yet read. To read it, you read port EB, where the received data byte is. If you read the UART status port you can detect an overrun, framing, or parity error by looking at bits D5, D4, and D3 respectively. If any of them are 1, the corresponding error has occurred. For example:

```
WAIT:   IN      A, (0EAH)     ; READ UART STATUS
        AND     10000000B     ; LOOK AT BIT 7-DATA REC'D?
        JR      Z,WAIT        ; NO DATA YET. WAIT.
        IN      A, (0EBH)     ; READ DATA.
        LD      C,A           ; SAVE DATA IN C
        IN      A, (0EAH)     ; UART STATUS-CHECK FOR ERROR
        AND     00111000B     ; MASK OUT ALL BUT ERROR BITS
        JR      Z,GOOD        ; GOOD, NO ERRORS.
        JP      ERROR         ; AN ERROR! PROCESS IT.
GOOD:   LD      A,C           ; MOVE GOOD BYTE BACK TO A
        CALL    DISPLY        ; DISPLAY REC'D BYTE.
```

In summary, the UART status port is used to detect the presence of received bytes, show any errors that may have occurred, and tell when a new byte can be sent out.

Data	*UART Status Register Bit Function*
3	Parity Error 1=True
4	Framing Error 1=True
5	Overrun Error 1=True
6	Transmitter Buffer Empty 1=True
7	Data Received 1=True

You should now have all the needed information to write software for your TRS-80 so that it can communicate with RS-232 devices. As an example of a simple application, here is a dumb terminal program. It initializes the UART from the switch settings, sends

out a byte each time a key is pressed, and displays each received byte on the screen. It ignores all errors.

A TERMINAL PROGRAM

```
DUMB TERMINAL PROGRAM–COPYRIGHT 1981, DAVID LINDBERGH
THIS IS A FULL DUPLEX DUMB TERMINAL PROGRAM FOR THE TRS-
80. IT SETS THE UART TO SWITCH SETTINGS AND IGNORES ERRORS.

LF       EQU    10            ; LINE FEED
CURON    EQU    14            ; CURSOR ON CHARACTER
KBD      EQU    2BH           ; GET KEY, NO WAIT.
DSP      EQU    33H           ; SEND A TO SCREEN
MRESET   EQU    0E8H          ; MASTER RESET PORT
MSR      EQU    0E8H          ; MODEM STATUS REGISTER
BRS      EQU    0E9H          ; BAUD RATE SELECT REGISTER
SWITCH   EQU    0E9H          ; SENSE SWITCHES
UCONT    EQU    0EAH          ; UART CONTROL REGISTER
USTAT    EQU    0EAH          ; UART STATUS REGISTER
UDATA    EQU    0EBH          ; UART DATA REGISTER

INITIALIZE UART TO SENSE SWITCHES

PROG:    OUT    (MRESET),A    ; DO MASTER RESET
         IN     A,(SWITCH)    ; READ SWITCHES
         AND    7             ; MASK OUT ALL BUT BAUD BITS
         LD     E,A           ; LOW BYTE OF DE HAS VALUE
         LD     D,0           ; HIGH BYTE OF DE IS 0
         LD     HL,TABLE      ; GET TABLE ADDR
         ADD    HL,DE         ; HL HAS ADDR OF CORRECT BYTE
         LD     A, (HL)       ; GET BYTE IN A
         OUT    (BRS),A       ; WRITE TO BAUD RATE SELECT
         IN     A, (SWITCH)   ; READ SWITCHES
         AND    0F8H          ; MASK OUT BAUD DATA
         OR     7             ; SET LOWER 7 BITS
         OUT    (UCONT),A     ; WRITE OUT TO CONTROL
```

UART IS NOW CONFIGURED. MAIN PROGRAM STARTS HERE.

```
        LD      A,CURON       ; GET CURSOR ON BYTE.
        CALL    DSP           ; TURN IT ON.
MAIN:   CALL    KBD           : GET 0 OR BYTE FROM KBD
        OR      A             ; IS IT A 0?
        JR      Z,RCHECK      ; IF YES, CHECK FOR R'D BYT.
        LD      C,A           ; SAVE A IN C
TWAIT:  IN      A,(USTAT)     ; READ UART STATUS
        AND     40H           ; LOOK AT BIT 6
        JR      Z,TWAIT       ; XMIT REG NOT EMPTY! WAIT.
        LD      A,C           ; GET BACK A FROM C
        OUT     (UDATA),A     ; SEND OUR BYTE
RCHECK: IN      A, (USTAT)    ; READ UART STATUS
```

Where Does It Go From Here?

In this chapter, we have looked at the input/output ports of computers and terminals to understand how the digital data is coded and what kind of signalling is typically used. Once the data is out of the computer, it must be changed into a form that can travel over telephone lines if it is going to be used for long-distance communication. The device that connects the computer to the phone line is called a modem. Modems are the subject of the next chapter.

4

The Mighty Modem

All data systems communicating over telephone lines need a very important piece of equipment to link the telephone lines to the computer or terminal. This device is called a *modulator/demodulator* or *modem.* The modem modulates and demodulates audio tones transmitted over the telephone circuits. The telephone system of the 1980s is designed to carry only audio signals, in the form of alternating current voltages. Computers and terminals transmit information in and out in the form of digitally-coded, direct-current voltages. At some time in the future, all communication systems may be digital, and the telephone systems may accept direct current signals from homes and offices, but until that time, we must feed sounds, not DC voltages, down the telephone lines. A modem sends and receives audio tones over the telephone line on one end, and direct-current digital pulses for computers and

terminals on the other end. The electronic components within the modem manufacture sound in response to digital voltage and digital voltage in response to sound.

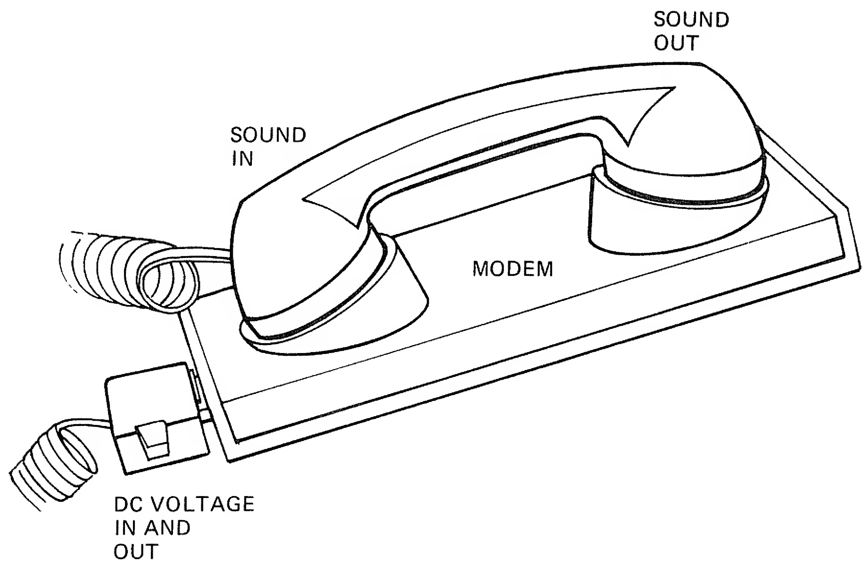


FIGURE 4-1. A modem and its functions.

SIGNALLING BY SOUND

The digital-voltage signalling standard between the modem and the computer or terminal is normally based on the EIA standard RS-232-C. The frequencies of the tones used over the telephone lines are determined by another standard. The most common tone signalling standard in the United States is the one established by Bell Telephone Laboratories called Bell 103. The Bell 103 standard is used for low-speed modems operating up to 300 baud.

In digital communication, we are only interested in representing the digital 0s and 1s. We need to show those two states or conditions only. If we use sound to transmit digital information, we could simply turn the sound off and on to represent 0s and 1s, but this off-on signalling would leave room for error. The absence of sound might be a correct signal, but it might also be a broken connection or an interruption by noise. Two-tone signalling was

adopted to represent digital information on audio systems. With two tones, we know that if tone A is not present, tone B should be. If neither tone is there, the system can immediately detect a problem.

If the Bell 103 standard, the station originating the call (usually thought of as a *terminal*) uses a tone of 2225 Hz to represent a 1 and a 2025-Hz tone to represent a 0. Obviously, if the modem at the answering end transmits back with the same two tones, only one side of the conversation will be able to "talk" at a time. If, however, the answering equipment uses two different tones, then we can use selective filters to detect only the desired signals and a simultaneous, two-way conversation can take place. Simultaneous, two-way transmission over telephone lines is referred to as *full-duplex transmission.* Most modems (and all the low-speed modems commonly used in electronic message systems) use four tones for signalling; two on the originate side and two on the answer side.

TABLE 4-1. Bell 103 standards

TYPE	*LOGIC*	*RS-232-C VOLTAGE*	*TONE*
Originate	1	–3 to –24 volts	2225Hz
Originate	0	+3 to +24 volts	2025Hz
Answer	1	–3 to –24 volts	1270Hz
Answer	0	+3 to +24 volts	1070 Hz

The tones listed in Table 4-1 are those transmitted by each side. A modem transmits one set of tones, but it receives the other. Many kit builders and home designers have forgotten this simple fact and ended up only receiving their own tones.

Higher-Speed Transmission

Other modem standards, such as Bell 202, Bell 212, and the signalling adopted by companies such as Anderson Jacobson and Racal-Vadic, are commonly found in commercial data communication systems. These modems use various schemes to provide higher signalling rates. The 202 series modems, for instance, are asynchronous devices that can transfer data at a maximum rate of about 1200 BPS on standard phone lines. But they only transmit

in one direction at a time *(half-duplex transmission).* They use a mark frequency of 1200 Hz and a space frequency of 2200 Hz. Complex *handshaking* signals are exchanged between the modems on each end of the line to control which modem will transmit at any given moment. The 202 signalling path sometimes includes a very slow (perhaps five baud) reverse channel over which the "nontransmitting" modem can signal for attention. This slow-speed channel is used to coordinate channel "turn-around."

The 202 signalling system is used commercially in systems where main computers and peripheral devices have lengthy blocks of data to send in one direction. They are commonly used in polling systems in which a computer contacts the peripheral device only periodically.

This signalling system is popular with amateur radio operators because it matches their typical half-duplex radio transmission, and the equipment is simple and relatively low in cost.

Bell 212

The more commonly used high-speed transmission method is Bell 212. Bell 212 modems operate in the full-duplex mode, but they do not use frequency-shifted keying. An originate modem operating in the 212 scheme transmits a 1200-Hz tone. The answer modem transmits a 2400-Hz tone. Each tone is shifted in phase instead of in frequency. A digital 1 or 0 transmitted at 1200 baud is only about .8 milliseconds long. This is not enough time to reliably detect the frequency of the tone. Changes in phase can be detected by electronic equipment very rapidly, but the phase-detection equipment is more sophisticated than a frequency-discriminating circuit. Phase-shifting modems can transmit data rapidly under full duplex, but they cost many times more than frequency-shifted devices. A commercial Bell 212 modem will cost between $750 and $1200 (Figure 4-2). They can be rented from most telephone companies for a monthly fee.

The Bell 212 and Bell 202 standard modems are not compatible with each other. The 212 standard is most commonly used by information utilities and other commercial services. The 202 standard is often used by private computer operators exchanging data with each other directly. It is conceivable that another hybrid standard may appear. A 202-style modem with a

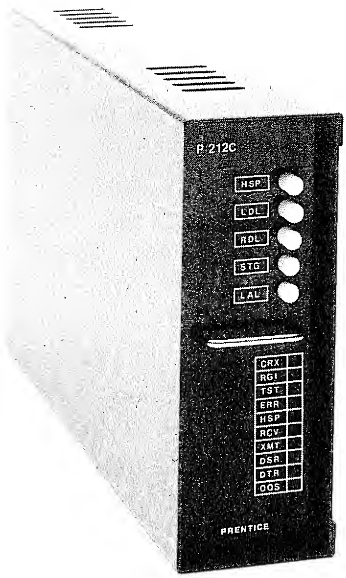


FIGURE 4-2. The Prentice P212C modem is a microprocessor-controlled device which can operate at 300 baud as a type 103 modem and at 1200 baud as a type 212 modem. It features sophisticated loop-back and testing modes and high reliability. It is priced over $800.

faster reverse channel would be perfect for communication between an information utility and a private user. If the reverse channel could operate at a minimum of 110 baud, it would meet the transmission needs of most users. The 1200-baud utility-to-user channel would speed the service and increase the enjoyment of information utilities.

Some information and message systems can provide high-speed transmission capability on an optional basis, but Bell 103 remains the standard for low-speed data transmission in the United States.

ORIGINATE AND ANSWER

The majority of 103-type modems have the ability to switch between either originate or answer tone sets, but some do not. The originate tone pair will be used by almost everyone using a microcomputer as a data communication terminal. The only two instances where the ability to transmit answer tones is necessary are: 1) when a computer is being used as an electronic message system, and 2) when two computer users are communicating with

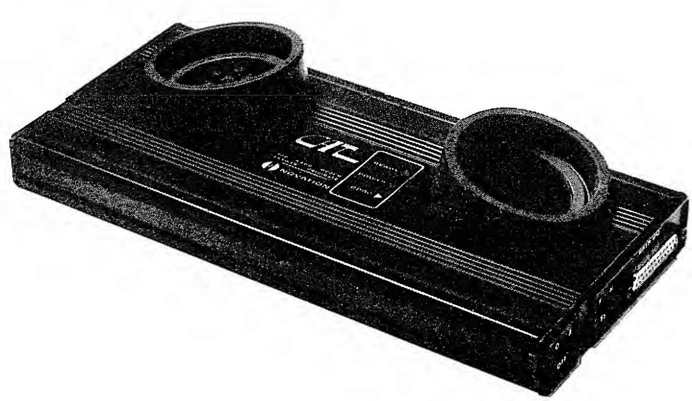
each other directly. In the first case, users of the system will be transmitting the originate tone set, and they expect to receive the answer tone set. In the second case, one of the operators must have a capability to transmit the answer tones in order to have a successful two-way simultaneous exchange. You might save some money by settling for an "originate only" modem, but be sure you will never need the answer capability. Most commercially available modems are now both originate and answer, but buyers should be particularly cautious when hunting for bargains. An "answer-only" modem would only be useful for special purposes.

The actual coding of the characters sent out by a modem is the same as the coding of the digital stream going in. If the computer's digital signals are ASCII-coded, then the tones will also represent ASCII characters. The coding of the characters remains the same; a modem only changes the signalling method from electrical pulses to audio tones.

A TYPICAL MODEM

Let us take a look at the actual installation and operation of a very popular modem, the CAT manufactured by Novation (Figure 4-3). The CAT is typical of many modems sold by Novation and other manufacturers; it serves as a good example of modem connection

FIGURE 4-3. The CAT modem, manufactured by Novation, Inc., is a popular device marketed by several different companies.



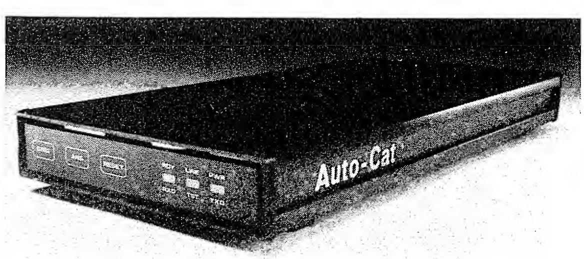
and operation. The CAT is sold under its own name, and at one time was sold under the Radio Shack label also. Radio Shack now markets a modem called Modem I that physically matches the TRS-80 Model I, II, and III computers. We will discuss the CAT because it has been so popular and because it illustrates the various connection options.

The CAT provides all of the standard features expected from a modern low-speed modem. It has a switchable originate-or-answer capability, a light-emitting diode that indicates a received signal, and reliable performance up to speeds of 300 baud. Included also is a self-test capability that allows the modem to listen to itself for testing purposes. This can be very handy when you are trying out new equipment or software.

DIRECT CONNECTION OR ACOUSTIC COUPLING

The CAT comes in two versions: direct-connection and acoustically-coupled. These terms refer to the way in which the modem marries to the telephone. A direct-connection modem actually plugs directly into the telephone system at some point (Figure 4-4). Some direct-connection modems plug into a wall jack, and others are inserted between the telephone instrument and the handset. The direct-connection version of the CAT is designed to be inserted between the telephone handset and the telephone body. The telephone is used in the usual manner, but when data

FIGURE 4-4. A more sophisticated version of the CAT modem provides for direct connection to the telephone line and automatic-answer capability. (Photo courtesy of Novation, Inc.)



communication is initiated, the handset is cut off and the modem electrically takes its place. This kind of connection means you must have a modern telephone with the handset attached to the body by small modular plugs and jacks. In the United States, you can now obtain updated telephones like these from local telephone companies, or you can purchase them from many retailers.

The modern telephone system is sensitive to certain tones used by telephone control and signalling equipment. Direct connection to the telephone lines is allowed by the Federal Communications Commission only under certain conditions. Direct-connection modems which plug directly into the wall must be certified to meet standards of tone purity and frequency. Uncertified equipment should never be directly connected to the phone lines. Telephone companies can provide devices known as data-access arrangements (DAA) to couple with uncertified equipment. Equipment connected on the handset side of the telephone instrument may not need certification, because the electrical network inside the telephone instrument provides the protection needed (Figure 4-5).

FIGURE 4-5. The MODEMPHONE by Racal-Vadic is a complete telephone and modem in one. The circuit board shown is installed in the MODEMPHONE. The DB-25 RS-232-C connector comes out the left side of the phone. Operation is controlled by the telephone cradle buttons and slide switch.

The second way in which modems can be connected to the phone lines is through acoustic coupling (Figure 4-6). In acoustic

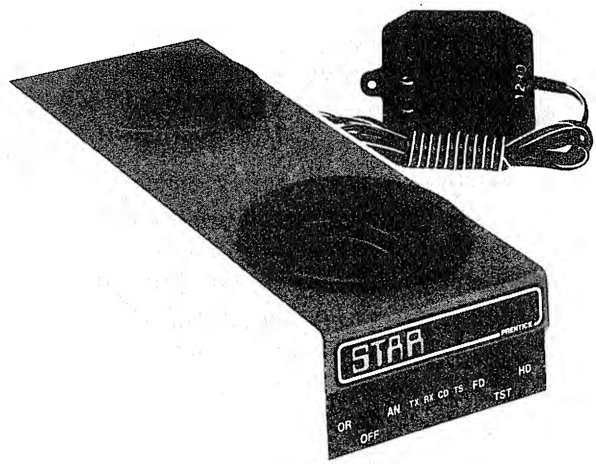


FIGURE 4-6. The Prentice STAR modem is a type 103 device equipped with very deep and secure acoustic cups. It provides reliable operation in a portable modem device.

coupling, the telephone handset is physically inserted into foam-rubber cups on the modem. Acoustic coupling has been used successfully for many years, and acoustically-coupled modems have been popular and successful. But acoustic coupling has only one advantage and several disadvantages.

The advantage of acoustic coupling is portability. If you intend to travel with your system, an acoustic coupler will provide the flexibility you need, as long as the places from which you communicate have telephones with fairly standard handsets. Modern streamlined telephones, some European telephones, and replicas of older telephones will not fit into the cups on most acoustic couplers.

The primary disadvantage of acoustic coupling is interference from outside noise. Noise inserted into the telephone from nearby persons, airplanes, radios, or any other source can disrupt and actually disconnect data communication.

Secondly, the mouthpiece of most telephone handsets is made from carbon granules which vibrate with sound waves and create electrical energy. When standard mouthpiece instruments are used for data transmissions over a period of many minutes, the carbon granules may "pack" or clump up and lose their activity. Tests have shown very large losses in efficiency after one hour's use. Packing takes place because the instrument is not being moved and because the constant repetition of the same two tones sets the granules into static patterns. Shaking or moving the handset can restore efficiency for a period of time.

Finally, some harmonics can be generated because of the physical structure of the handset itself. These harmonics can be degrading to a signal trying to get through a weak or noisy connection. Novation and other dealers also sell dynamic or condenser microphone elements which acoustically-coupled modem users can quickly insert into any standard handset. These elements are inexpensive, do not pack or lose efficiency, and limit harmonic distortion. They are very effective in increasing the reliability of acoustic modems. Novation calls their condenser-type microphone with electronic pre-amplifier the Super Mike (Figure 4-7).

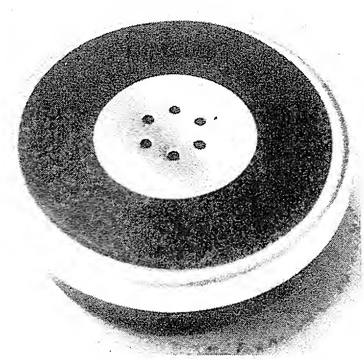


FIGURE 4-7. The Super Mike from Novation replaces the mouthpiece in most telephones and provides distortion-free transmission of modem tones when acoustically coupled modems are used.

THE DC SIDE OF THE MODEM

So far, we have discussed the telephone end of a modem. The end of the modem connecting to the computer or terminal is more complex and requires more attention to details.

Some modems come with a cable already made for the system with which they will be used. Radio Shack supplies prewired cables for their computer systems. But even if you have a prewired cable, it is helpful to know what signals are passing over the wires. If you do not have a prewired cable, there are some things you need to know to keep the 1s and 0s flowing in the right direction. The following description applies to all standard modems like the CAT and Modem I.

WHAT SIGNAL GOES ON WHAT WIRE?

The connector normally used on RS-232-C cables is called a DB-25. A DB-25 has 25 pins, and almost every one of them can be used for something in the RS-232-C signalling scheme. The wiring standard was originally designed to connect a terminal to a communication device. The pins are usually identified from the point of view of a terminal. However, in modern practice, some terminals are wired as communication devices because that is the role they serve (Figure 4-8). Many terminals have switch or wiring options. If you are connecting a modem to a terminal wired as a communication device, the wiring standard can become very confusing. Remember, there are communication devices and there are data terminals. A computer is a data terminal, but a commercial terminal may be wired either as a data terminal or a communication device. It is really not as hard as it sounds. Let us look at an example.

Although the connectors have 25 pins, the cable connecting a modem to a computer may have as few as three wires. One wire must be used to get data from the computer to the modem. Looking at the DC Cable Wiring Chart, we can see this wire is connected to pin 3 of both cable plugs. Another wire has to be used to get data from the modem to the computer. This is connected to pin 2. Finally, we need a common signal ground wire to complete both direct current paths. This common ground is always carried on pin 7. There is one other connection we must make on the plug at the computer end. Since our transmissions are on a full-duplex circuit, we need to defeat a signalling option provided for half-duplex transmission.

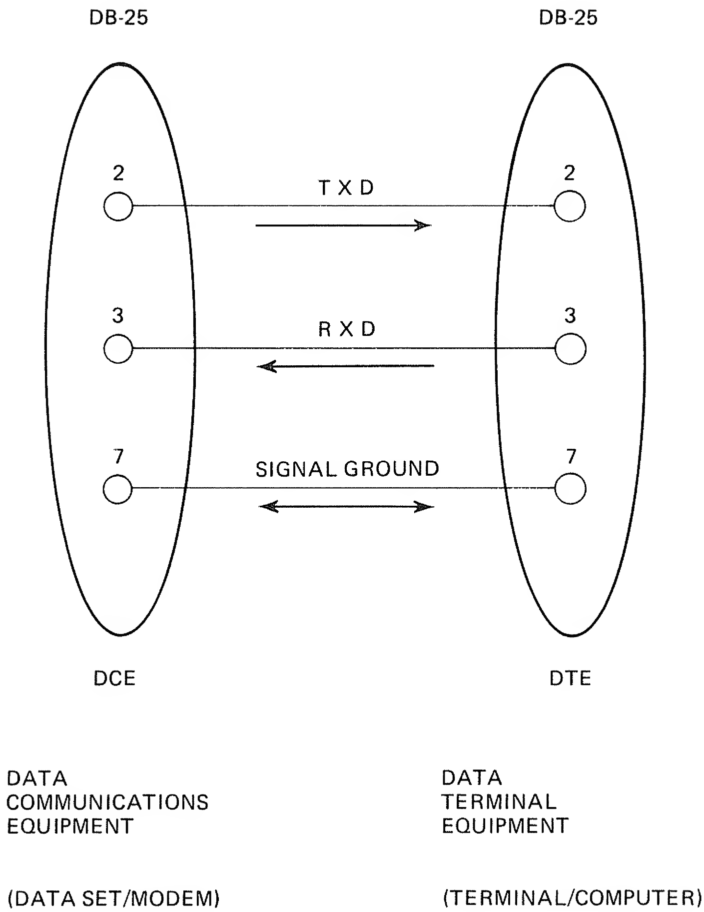


FIGURE 4-8. DTE/DCE connections.

In half-duplex transmission, the devices on each end of the line pass signals called request to send (RTS) and clear to send (CTS) to coordinate their data bursts. Since we use a Bell 103 full-duplex modem system, the RTS and CTS signals are not needed. The easiest way to bypass this option is to route the computer's own clear signal back to its request line. This involves putting a short jumper between pins 4 and 5 of the DB-25 plug attached to the computer.

The wiring of a modem to a computer is pretty simple compared to the case of a modem being attached to a terminal set up as a communication device. Both devices follow the communica-

tion equipment terminal-wiring pattern. Both devices want to receive data on pin 3 and transmit it on pin 2. Obviously, hooking two transmits and two receives together results in no data transfer. Here again, we must fool the system. We must transpose the wires in the cable so that pin 2 of each connector is attached to pin three at the other end. It seems obvious once it is explained, but many people have spent hours wondering why their modem will work with a computer acting as a terminal, but not with another terminal device. (See Table 4-2.)

TABLE 4-2. RS-232-C Wiring Guide

Pin	*Mnemonic*	*Function*	*RS-232-C Designation*
1	GND	Protective ground—May tie together chassis so all equipment is at the same ground potential.	AA
2	TXD	Transmit Data—Carries data from the modem to the computer.	BA
3	RXD	Receive Data—Carries data from the computer to the modem.	BB
4	RTS	Request to Send—Controls the transmission on half-duplex circuits. RTS ON, plus 3 volts or more, tells a modem to transmit.	CA
5	CTS	Clear to Send—Modem response to an RTS signal.	CB
6	DSR	Data set ready—Modem says it is on and ready.	CC
7	GND	Signal ground—MUST be used.	AB
8	DCD	Carrier detect—Modem indicates that it hears a tone.	CF
9, 10		Used for special testing purposes.	
11		Not used.	
12		Secondary received line signal detector—May be used for a second, slow-speed circuit.	SCF

TABLE 4-2. RS-232-C Wiring Guide **(continued)**

Pin	*Mnemonic*	*Function*	*RS-232-C Designation*
13		Secondary clear to send–May be used for a second, slow-speed circuit.	SCB
14		Secondary transmitted data–May be used for a second, slow-speed circuit.	SBA
15		Transmission signal element timing–For synchronous data transmission.	DB
16		Secondary received data. May be used for a second, slow-speed circuit.	SBB
17		Receiver signal timing–Used in synchronous data transmission.	DD
18		Not used.	
19		Secondary request to send–May be used for second circuit.	SCA
20	DTR	Data terminal ready–Computer tells auto-answer modem it can answer the phone.	CD
21		Signal quality detector–Used by special equipment.	CG
22		Ring indicator–Modem says the phone is ringing.	CE
23		Data signal rate selector–Not used in serial ports.	CH/CI
24	TXC	Transmit clock–Used in synchronous transmission.	DA
25		Not used.	

The standard uses the phrases Data Communications Equipment (DCE) and Data Terminal Equipment (DTE). DCE obviously includes modems, but sometimes printing and video terminals are wired as DCE devices. DTE always includes computers, but some-

times terminals are wired as data terminal equipment. This can cause some confusion in how pins 2 and 3 are used. Data communication equipment is equipped with a female DB-25 connector. Data terminal equipment usually has a male connector.

A control circuit is considered ON when the voltage is more positive than +3 volts. It is OFF when the voltage is more negative than −3 volts.

FULL- AND HALF-DUPLEX MODEMS

The term full-duplex (FDX) usually means that data can be sent and received simultaneously by a data communication device. In modems, that is exactly what the term means. The terms full- and half-duplex take on a slightly different meaning in reference to terminals, and we will discuss those shortly.

With modems, full-duplex means simultaneous bidirectional communication using the two sets of modem tones described earlier. This is the most common form of operation for low-speed data communication systems. Half-duplex means that the modem can only transmit in one direction at a time. This is similar to ham radio transmissions where each operator says "over" at the end of a transmission to signal the other that it is his turn. Half-duplex operator requires that the terminal and computer control the modems through the "request to send" lead in the RS-232-C cable. Additionally, a modem operating in half-duplex will echo back a terminal or computer's own characters back to it—a job usually done by the receiving end in a full-duplex operation. Half-duplex may be used in some unique situations, that is, where transmission in one direction is made at a slower speed because of the limitations of a printer on the line, but full-duplex is the standard mode of operation for modems used with information and message systems. Some modems have a full- and half-duplex switch. Modem I does not. It always operates as a full-duplex modem except in one special case.

The Modem I can be used with a TRS-80 Mode I without the Expansion Interface or serial card. Modem I is unique in that with a special cable (catalog number 26-3009) and software (26-1139), it can be connected to the TRS-80 Model I's cassette port.

But it will only operate in the half-duplex mode. This is not a major drawback since most message systems and information utilities can be commanded to operate with a half-duplex terminal. Let us see what that means.

FULL- AND HALF-DUPLEX TERMINALS

The terms full- and half-duplex take on slightly different meanings when they are associated with terminals or computers acting as terminals. A terminal operating in half-duplex displays its own characters on the screen (or printer) as they are transmitted. A terminal operating in full-duplex (or echo-plex as it is sometimes called) expects to have its transmitted characters echoed back from the distant end (or from its own modem if the modem is in half-duplex). In turn, it echoes back characters it receives. A distant end echo serves as a positive and constant check of the quality of the transmission path (Figure 4-9). If you are operating

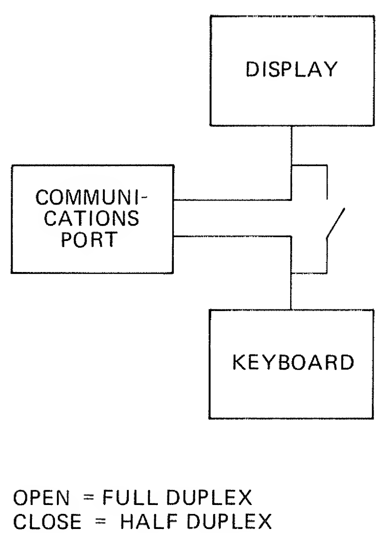


FIGURE 4-9. Full- and Half-Duplex Terminal.

in full-duplex and see garble coming up on your screen, you know something is wrong in the signal path. You can't be sure if the problem is on the transmission from you to the distant end or from the distant end to you, but you can be sure a problem exists.

As handy as full-duplex transmission is to insure circuit quality, it can also be a source of confusion to inexperienced operators finding themselves with two or more characters on the screen for every one they type. If your terminal is in half-duplex and your modem is in half-duplex, your terminal will display a character as it is being sent and again as the modem echoes it back. If the other end of the circuit is connected and operating in the full-duplex mode, it also will echo the character back and you will get three characters on your screen every time you touch one key. This can be a real surprise, particularly to a speedy touch typist. The product on the screen is a jumble of characters and echoes of characters, but luckily the cure is simple. Just read the simplex/duplex chart and set up your system for the proper mode of operation. The "full-half" switch on the CAT and similar modems should normally be left in the "F" position except when you are running a test where you want a local echo of your own character transmissions (Table 4-3).

TABLE 4-3. Full/Half-Duplex Settings

SYSTEM A		*SYSTEM B*	
CASE 1			
MODEM	FULL-DUPLEX	MODEM	FULL-DUPLEX
TERMINAL	FULL-DUPLEX	TERMINAL	FULL-DUPLEX
CASE 2			
MODEM	HALF-DUPLEX	MODEM	HALF-DUPLEX
TERMINAL	FULL-DUPLEX	TERMINAL	FULL-DUPLEX
CASE 3			
MODEM	FULL-DUPLEX	MODEM	FULL-DUPLEX
TERMINAL	HALF-DUPLEX	TERMINAL	HALF-DUPLEX

Only case 1 will allow full echoing by both systems to provide a constant check on the quality of the circuit. In case 2, the modems each will echo back to their respective terminals. This only provides a check of the local modem/terminal circuit. The configuration in case 3 is essentially no echo with the local display showing what was entered on the local keyboard.

Any other switch or logic combination will result in either multiple characters or no characters being displayed.

SELF-TEST

The CAT provides a self-test feature which is great for trying out terminal software and hardware operation before actually going on-line with a remote system. The test mode is designed to check all portions of the modem's operation. It does this by making its own transmit tones the same as the receive tones. It still needs a telephone set in place to feed the tones from the transmitter to the receiver. The telephone has to be free of any dial tones or ringing signals. You can often achieve this for 30 seconds or so by dialing the first digit of your local exchange. This test feature should provide you with a complete acoustically-coupled check of your entire system. It should echo everything you send. Modems without the test feature can provide an echo check when operated in the half-duplex mode, but this does not verify the operation of the audio portions of the modem.

After a successful self-test, you are ready to go on-line with a host system or a friend. The combination of cabling, echo, and tone options may seem confusing at first, but they seldom change in operation and soon become invisible to your enjoyment of the world of data communication.

Modems like the CAT and Modem I are fine cost-effective devices for most communication situations where a serial output is available from the computer. But often a serial output is an expensive option. Some modems are available which connect directly to the computer's main data path.

MODEMS FOR COMPUTERS WITHOUT A SERIAL PORT

Several manufacturers have developed modem devices which plug directly into the main circuit board of the computer and interface the parallel data bus they find there with the serial outside world. The LYNX and Microconnection are two such devices for the TRS-80 Model I and Model III.

A "bus-decoding" modem like the LYNX or Microconnection comes equipped with a cable that allows it to plug directly into the TRS-80 without the use of a serial interface card or ex-

pansion module. Versions of both devices are available for both Model I and Model III machines.

The LYNX requires no unique software. It will operate with all standard terminal programs. It is strictly a modem device. It converts the parallel data into audio tones with no other outputs or inputs in between.

Because of unique interfacing, the Microconnection requires communication software with special address locations. The Microperipheral Corporation supplies appropriate software with each system, and their smart terminal program is very good. In addition, several sources of communication software sell versions especially adapted for use with the Microconnection.

The bus-decoding devices in the Microconnection series have a unique capability in that they will not only interface with its appropriate computer and with the telephone line, but they also have an RS-232-C connector built in which can serve as a complete serial port in itself. The RS-232-C port can be used in several ways. It can be used to feed a printer which can print along with the data passing in or out of the modem in order to provide a permanent copy of data transmitted or received. It can be used as a separate RS-232-C connection for the computer without the modem/telephone portion of the system being activated. Finally, a bus-decoding Microconnection can serve as a "standard" RS-232-C modem without the use of the special interface to the computer system. The trade-off for this flexibility is the need for software to move the data in and out of the Microconnection.

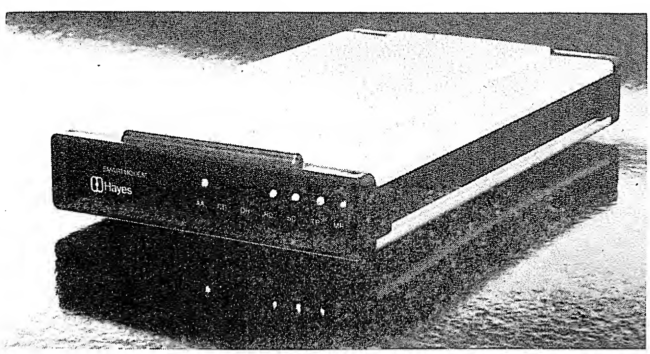
An advantage of some modem devices interfacing directly with the internal computer data stream is the ability to dial a telephone number automatically in response to commands from the keyboard or from software. These devices also provide an automatic answer capability which, combined with the right software, permits unattended remote access of the host computer. These features are standard on some devices and optional on others. All directly interfacing modems are also direct-connection modems; they do not use acoustic coupling. The Microconnection has another option for the use of European tone standards.

Modems connecting directly to the computer's data bus structure can provide a data communication capability at a reduced price. The decision to purchase the more traditional serial port/modem configuration and the bus-connected device depends

a great deal on what equipment is already available and what special operating features you may desire. Later chapters in this book will describe more of the options you may wish to explore to increase the value of data communication to you (Figure 4-10).

FIGURE 4-10. The Hayes Stack Smartmodem is an "intelligent modem" containing it's own microprocessor and operating system. The intelligent modem connects to a computer or terminal through an RS-232-C port. It receives the data to be transmitted and the modem operating instructions over the same data line. The modem's microprocessor monitors the data stream for its own unique command line. It has complete auto-answer and auto-dial capabilities. All operating parameters can be controlled from the keyboard.

The Hayes Stack Smartmodem provides unique flexibility because the computer or terminal it is working with does not need to have any interfacing software. The modem is so "smart" that the terminal can be very "dumb." This device sells for well under $300 and accessories such as a clock console are available.



And Now. . . ?

We have examined ones and zeros; ASCII, RE-232-C, and Bell 103; modulators and cables. In the next chapter, we will look at more practical operational features. We will try to answer the question, "What is a terminal, and how do you tell if yours is dumb, smart, or brilliant?"

5

Terminals

Terminals are the doorways of a computer system. They let your data in and out. Some move from side to side, and others roll up and down. Some function automatically, and others are just plain dumb doors. But without them, the world of data communication would be closed off to individual entry.

Terminals come in every size and shape. The cash register at your corner grocery store may be a terminal for a large inventory and billing system. Employees performing an inventory of the grocer's shelves may enter data into terminals that look like hand calculators. Micro- and minicomputer can serve as terminals for much larger computer systems. The devices that print airline tickets are actually receive-only terminals. Most of us have seen video terminals in use at airports and municipal offices. Modern industry, travel, and education are all becoming increasingly tied to computer terminals.

TERMINAL PARTS

Every terminal has three basic elements: an input device, a display device, and a communication port. The input device may be a full keyboard, a limited calculator-like keyboard, or devices such as light pens and scanning wands now found in many stores. Paper tape and punch cards have long served to feed data into terminals equipped to read them.

Display devices include various kinds of printers, cathode ray tubes (CRT), flat-faced video screens called plasma displays, and paper tape and cards. The graphic scoreboards found in sports arenas are actually the display devices of specialized computer systems. The terminal in your home, school, or office will be smaller than the one in the stadium, but it can be just as spectacular.

The communication port on a terminal may use any of the signalling or alphabet schemes described in previous chapters. The RS-232-C/ASCII configuration is the one most commonly used outside of IBM systems. Most video terminals have two RS-232-C ports. One port is used for communication and the other is used to drive a printer. In this way, a paper copy can be made of data entering or leaving the video terminal.

PRINTING TERMINALS

There are several different types of terminals used in data communications systems. Originally, most terminals were teleprinters and automated typewriters. Many printing terminals are still in use (Figure 5-1). These printers give a paper record of everything transmitted and received and have many important applications. Printing terminals may have some speed restrictions. The mechanical printing action may not follow the fastest communication circuits. Printing terminals can consume large quantities of paper and are usually noisy, but often there is no substitute for having a paper copy of the communication interchange. Until recently, printing terminals had a corner on the portable terminal market. The printing mechanism could be made much more portable than a cathode ray tube-based system. This has changed with the introduction of pocket-sized terminals using liquid crystal displays.

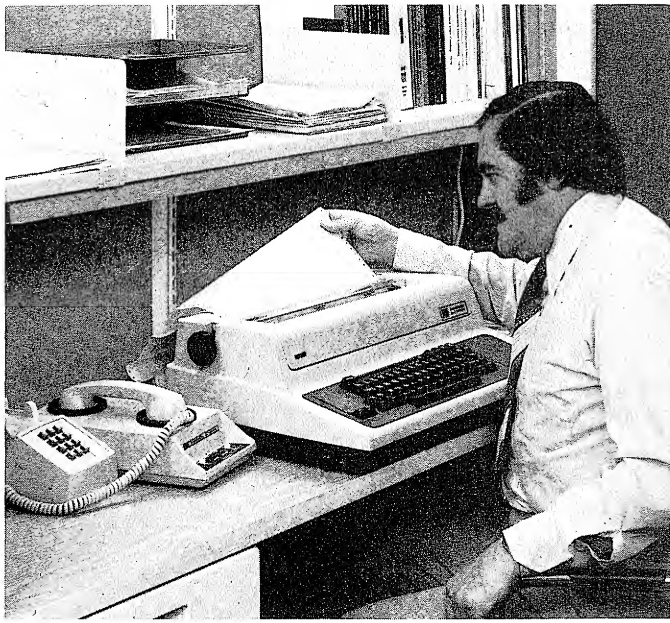


FIGURE 5-1. Printing terminals can be very convenient for many applications, particularly where extensive editing is not required. The Anderson Jacobson AJ 830 terminal and AJ 247 data coupler/modem make a practical and efficient work station. (Photo courtesy of Anderson Jacobson.)

CRT systems form a separate branch of the terminal family called video terminals.

VIDEO TERMINALS

Video terminals are made by many different manufacturers and come with many different features and options. These options can be divided into three categories: screen options (including size, display, cursor control, and others), keyboard options, and "intelligence."

Intelligence is defined as the amount of electronic help built into the terminal device. Using the analogy of doors, a "dumb" door is one that you have to unlock, turn the knob, and open. A "smart" door is one that unlocks and opens in response to a button you press. A "brilliant" door recognizes you when you

walk up and opens automatically. Terminals come in about those same three categories. If you operate a microcomputer as a terminal, you may operate in either the dumb, smart, or brilliant modes. After we examine some standard terminal options, we will move on to the nuts and bolts of making a microcomputer into a terminal device.

Screen Options

The screen options available in terminals relate to how the display looks and what you can do with it. Some terminals can provide multicolor displays of graphic characters. Most provide a white-on-black screen display. *Reverse video* (black letters against white) and *highlighting* are common options. Simple charts or bar graphs can be displayed using standard characters to build the lines of the chart. Screen sizes range from a small 5-inch diagonal to more standard displays of about 12 to 14 inches. Two important screen functions are *formatting* and *cursor control.*

Formatting refers to how blocks of letters and numbers are organized on the screen. Some terminals allow information to be displayed on *split screens*—like the pages of this book. Others provide *status lines* which are fixed lines of internal information about the transmission mode selected, characters sent and received, and other options selected.

The cursor on a terminal is a visual reference point. It usually shows where data will next be entered. Some cursors are simple blocks of light. Others are special symbols which change or blink off and on when certain options are selected. Many terminals can only enter data across the page and line by line—just like a typewriter. Others allow their cursors to be "flown" around the screen so that data can be entered or corrected at any spot on the "page." Cursors can often be moved by using directional keys, and some units allow positioning of the cursor on the screen by using numerical **X** and **Y** coordinates. Other cursor options include backspacing and an automatic "home" position at the top left corner of the screen.

Line width is an important variable. This is not related to physical screen size, but rather to the number of characters displayed in a line. Most standard terminals will allow up to 80 characters to be printed across in a line. (This is a carry-over from the 80-column punch card.) Many will display only 64 characters,

but this is standard letter width so it is perfectly acceptable for many activities. Most terminal screens allow for the display of at least 24 lines of data.

Keyboard Options

The keyboards found on video terminal devices may range from austers models with about 50 keys to giants with over 100 keys and switches. The ASCII alphabet contains 128 characters and codes. A terminal keyboard that does not have the ability to transmit all of those codes may be limited in some applications.

Many computers will only recognize upper-case letters, so most terminals have a standard "uppercase only" mode of operation available. However, when sending electronic mail, it is often nice to use the common upper-case/lower-case mix. Many terminals feature a separate numeric pad for frequent entry of number data. Other special feature keys include: cursor control, tabulation, backspace, and screen formats (for example, split screen). Many terminals include programmable special function keys. These keys perform certain specially-defined functions when the terminal is running under the control of an internal program. Possession of an internal operating program and internal memory puts a terminal into the intelligent category (Figure 5-2).

FIGURE 5-2. The RCA VP 3303 Data Terminal is a unique device in many ways. It is designed for use with information utilities and Videotext systems. It features a full keyboard, which is rugged and water resistant. It can use a home television set or monitor to display color and graphic screens of information and data. The system includes sound effects and many other operating features.

TRS-80 owners could use the VP 3303 as a color output device by writing programs which send the terminal specific ASCII characters over the RS-232-C line. The terminal can translate the normal ASCII characters into colorful graphics. This terminal is a powerful microprocessor-controlled communications device that can convey information in more ways than just the printed word.

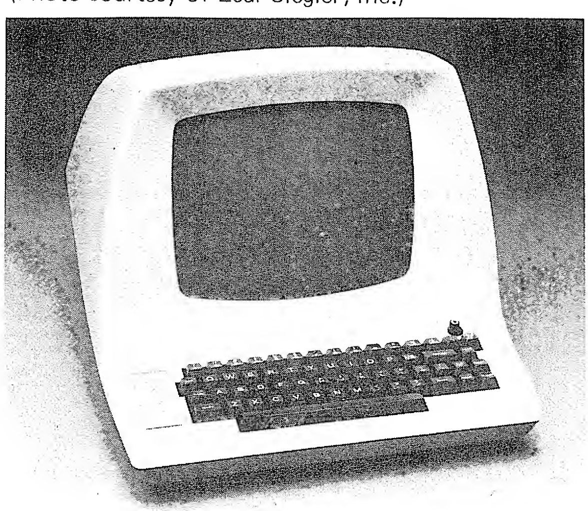
Dumb, Smart, and Brilliant

Dumb, smart, brilliant—how can a terminal device be described by these words? Terminals, like some people, can be classed by how useful they are in doing the really hard or detailed jobs. These classifications depend upon the degree of programming and memory available within the terminal device.

A *dumb* terminal is nothing much more than a keyboard, serial port, and display screen. The keyboard may be limited in the characters it can transmit. Often it cannot transmit lowercase characters or some control codes. The cursor simply marches along like a typewriter. It cannot be positioned on the screen. When data comes in from the device on the other end of the communications link, it is printed out on the screen from top to bottom. When the screen is full, the text scrolls off the top and is lost.

During transmission, data to be transmitted must be typed in a character at a time while the terminal is connected to the distant device. All access codes, passwords, and other repetitive material must be typed in each time it is used (Figure 5-3).

FIGURE 5-3. The Lear Siegler ADM-3A is the device which first introduced the concept of a "Dumb Terminal." In fact, Dumb Terminal is their trademarked term. It has been the standard terminal against which many others have been measured. (Photo courtesy of Lear Siegler, Inc.)

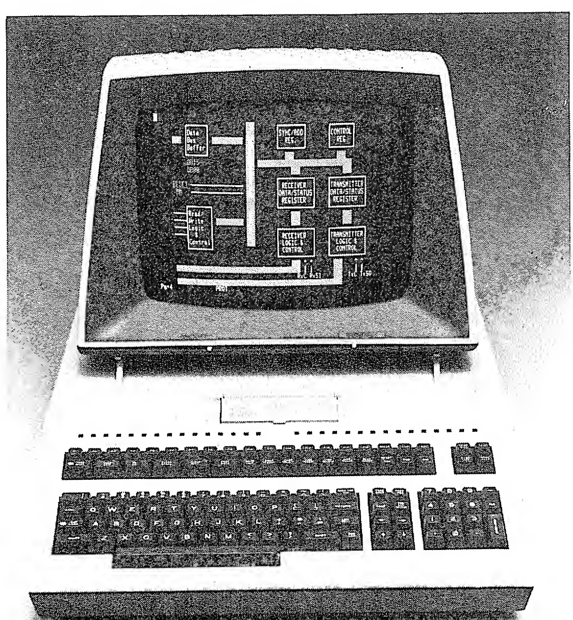


Dumb terminals are sufficient for many uses and relatively inexpensive. They are often found in schools and in applications where neither operator nor computer time is at a premium. Adding some internal memory and programming to a terminal brings it into the *intelligent* class. The first step in the intelligence scale is *smart*.

Perhaps the definitions of *intelligence* and *smart* should be caveated by adding that different manufacturers define the terms to fit their equipment line. Let's look at one terminal that calls itself *smart* and see what features are provided.

The Lear Siegler ADM-42 is called a "powerful smart terminal" by its manufacturer (Figure 5-4 and 5-5). It has many of the screen and keyboard options discussed earlier, including the ability to display characters to attract attention, highlighting, a fixed line showing the status of the terminal, and various editing commands.

FIGURE 5-4. The Lear Siegler ADM-42, like the Hazeltine Modular One, is a powerful terminal. It, too, has a graphics capability, full cursor control, detached keyboard, special function keys, pages of internal memory, and many other features. (Photo courtesy of Lear Siegler, Inc.)



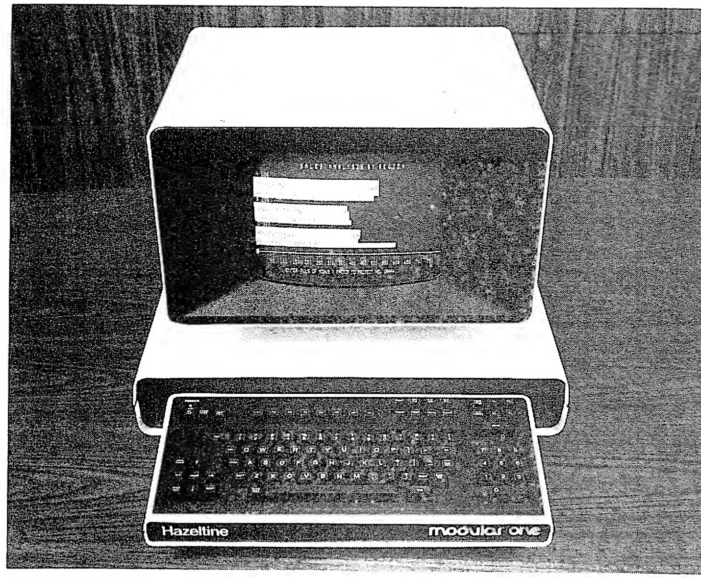


FIGURE 5-5. The Hazeltine Modular One is a powerful terminal providing many operating conveniences. It has a graphics capability, full cursor control, detached keyboard, special function keys, pages of internal memory, and many other features. (Photo courtesy of the Hazeltine Corporation.)

It has an internal memory which can hold up to eight pages of text (2000 characters per page). It has 16 special function keys which can transmit 32 prestored messages with one keystroke. These messages can be as simple as one or two ASCII characters plus a carriage return or as long as 63 characters.

In operation, these features can provide increased efficiency for both the operator and the computer. The operator can compose up to eight pages of data (with the maximum ADM-42 memory option), hold them in the terminal, and edit them until they are perfect. Only then will contact be made with the computer so the pages can be transmitted efficiently at high speed. This method of composing data for entry saves on communications charges and computer time.

When entry is desired into the distant end or host computer system, the special function keys can help to automatically open the door. Computers usually require very specific log-on and entry sequences. Each step in the entry sequence can be accomplished

by a pre-set special function key. Computer entry can be as simple as pushing keys 1, 2, and 3. Similarly, many data and text entry functions require special commands. Programming these into the specially marked function keys ends the hunt for lost pieces of note paper and time consuming reference to operating manuals.

When data is received from a distant or host system, the pages of memory available in most smart terminals can allow the operator to review the material off line, re-edit it, and then print it out on a printer attached to the second RS-232-C port. This ability saves operator time, computer time, and paper handling.

A *brilliant* terminal is actually only an extension of a smart terminal's capabilities. A brilliant terminal has more internal memory, external memory for long-term storage, and its own operating programs. In short, this is a job for a fully equipped microcomputer.

The Perkin-Elmer 3500. If we took all the features and functions of a smart terminal and added a 6800 microprocessor, 48K of RAM, 2 disk drives, and an extensive package of utility and communications programs in ROM *(read only memory)* we would truly have a versatile communication machine. That description fits the Perkin-Elmer model 3500 Intelligent Terminal (Figure 5-6).

The 3500 is a microcomputer system optimized to perform as a terminal. It has an extensive disk-operating system in ROM and comes with BASIC and assembly language programming features. As a terminal, it has every keyboard and screen feature imaginable. It can display selected characters at half-intensity, blinking, black on white, or underlined. It has special characters for business forms and provides movement of blocks, lines, and columns of data at the stroke of a key.

As each screen of information is developed, it is compressed and saved on disk by the operating system. The disk can provide long-term storage of all data transmitted and received by the terminal. The operating system provides all of the software needed to process disk information locally or transmit it to another system or terminal.

Because this terminal can be programmed with detailed instruction, it is capable of completely unattended operation. Using an automatic dial modem, it can call a host system, sign-on, cue the host for the activity desired, transfer data, and termi-

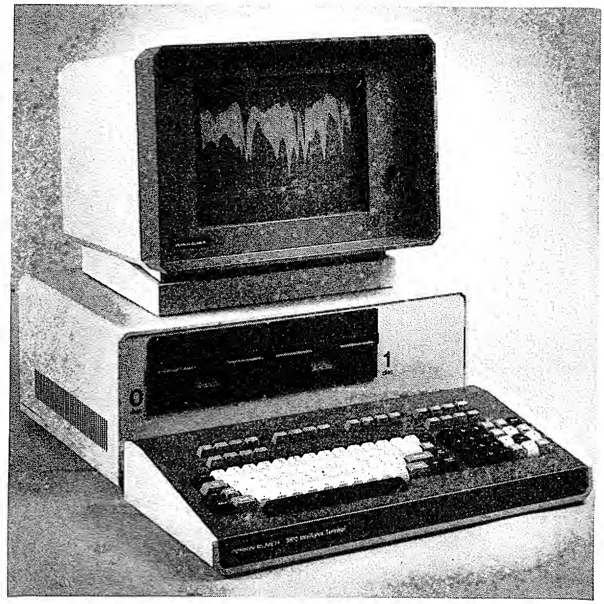


FIGURE 5-6. The Perkin-Elmer 3500 Intelligent Terminal is actually a stand-alone computer with specially developed capabilities for operation as a terminal. It can manipulate data in disk files locally and exchange data with larger mainframe systems. (Photo courtesy of Perkin-Elmer.)

nate the call—all without an operator. It can do this late at night when communication costs are low and equipment is not busy.

The Perkin-Elmer 3500 evolved from a terminal into a microcomputer specialized for data communication. It is possible to approach the situation from the other way and specialize a standard microcomputer to act as a terminal. You may not get all of the sophisticated videos and features of the 3500, but you can develop a brilliant and custom-tailored operating system.

PORTABLE TERMINALS

Before we leave our overview of dedicated terminal systems, we should focus on the smallest members of the terminal family, the portable terminals. Portable terminals help remove an obstacle to communication I call the "Limitation of Location." The Limita-

tion of Location used to mean a data communication user had to physically be near a rather bulky terminal device in order to communicate. The hand-sized portable terminals now on the market make it possible to duck into a phone booth and make a quick data call as easily as a voice call can be made.

In 1967, MSI Data Corporation introduced a cart-mounted modified adding machine and tape recorder powered by an automobile battery that allowed a supermarket clerk to order products by entering data via the keyboard onto magnetic tape; the data was then transmitted to the warehouse over telephone lines. In contrast, the newest portable terminal weighs 22 ounces, is powered by four penlight batteries, contains a microprocessor and up to 64K of memory, and can be programmed for a wide variety of applications by the user. The MSI Omega Generation series of portable terminals is available with various programming and communications options. The terminals are used in areas such as inventory, field service calls, utility meter reading, and manufacturing production lines (Figure 5-7).

RCA markets a hardy and portable terminal called the VP-3501 which includes a full function keyboard and numeric pad and a built-in modem all in one moisture resistant case. It can interface with any television set and provide color graphics display. This full capability terminal weighs under 5 pounds (Figure 5-8).

FIGURE 5-7. The MSI Data Corporation MSI 88f is a programmable portable terminal able to provide a handful of sophisticated operations. (Photo courtesy of MSI Data Corporation.)

FIGURE 5-8. The RCA VP-3501 portable terminal includes an integrated modem and video modulator to allow direct connection to any television set.

In operation, some allowances have to be made for the keyboard and display size of any portable terminal. Typically, each key has several functions it can perform. Operators have to learn specific shift functions to get the most out of each keystroke. Numbers can be entered quickly, but typing letters is slow until skill is gained through practice. These units display only a single short line at a time, but data can be stored and replayed line by line if needed. Dedicated host systems can easily be programmed to transmit data line by line when working with portable terminals. A separate modem device is needed to communicate over telephone lines. An acoustically-coupled or direct-connection modem can be used.

The most common function for a portable terminal is data collection in the field. After the data is collected, it is held and later dumped to the host system directly or by telephone (Figure 5-9). The Limitation of Location has been severely bent, if not broken.

A very sophisticated portable terminal device such as the MBI 88e is capable of all of the program options found on any larger intelligent terminal—including pre-stored log-on and transmission

FIGURE 5-9. Portable terminals are becoming much more important for the collection of data away from traditional computer work-station locations. This utility worker is entering data that will later be transferred to a mainframe computer system for analysis. (Photo courtesy of MSI Data Corporation.)

prompts. This hand-held terminal is limited in size only by the need to make the display and keyboard easy to use.

Enter Through the Door

In the beginning of this chapter, terminals were described as doorways to data communication systems. We have seen many kinds of doors (dumb, intelligent, and portable), but the purpose of the devices has always been to let the data flow in and out of communicating systems. In the next chapters, we will look at practical ways in which available microcomputer systems can be made to serve as data communication devices.

6

Using the TRS-80 as a Terminal

The Radio Shack TRS-80 line of computers, particularly the system now called the Model I, has become the most popular microcomputer system in the world. My survey of thousands of information utility users showed that 31.5% were using TRS-80s to communicate. The Radio Shack line of systems is extensive and the number of options and combinations of accessories from many manufacturers is very large. This chapter will concentrate mainly on untangling the world of data communication options for the TRS-80 Model I and III computer systems. The Model II and Color Computers have very simple and efficient communication capabilities that are easy to use and explain. Let's take the biggest bite first and sink our teeth into the TRS-80 Model I.

THE MODEL I

The TRS-80 Model I is no longer being made. It was the victim primarily of increased regulation on the amount of radio frequency interference microcomputers are allowed to emit. However, with well over 250,000 units sold, this system will be important to anyone interested in microcomputers for years to come (Figure 6-1).

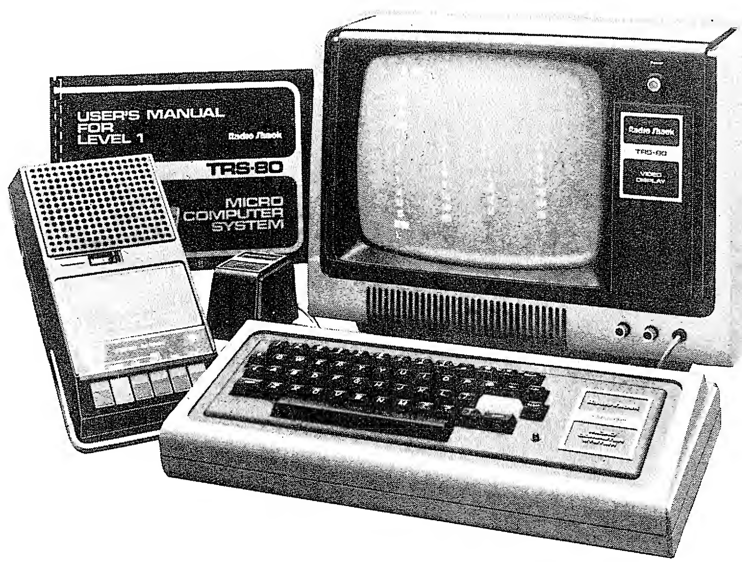


FIGURE 6-1. The TRS-80 Model I is no longer being sold, but it is still in use around the world. The addition of the proper accessories can make it into a very valuable data-communications device.

It is important to understand the components of the TRS-80 Model I system so the role of some of the alternative pieces of equipment makes sense later. The "keyboard" section houses the actual computer CPU, ROMs for the operating system and the BASIC language, and up to 16 K bytes of memory. Various connectors on the back of the keyboard bring in power, the cassette I/O port, and the expansion bus. The expansion bus on a TRS-80 Model I is extended through a 40-pin connector and cable. The

system enclosed in the keyboard housing is the fundamental building block of the TRS-80. It is a stand-alone computer capable of loading and saving programs for cassettes and running complex BASIC programs in the RAM space available. Many manufacturers have provided options to hang on the system from this point, but let's see how it was originally designed to be expanded.

The Expansion Interface for the TRS-80 Model I provides room to house extra RAM, a controller chip for the disk system, a real time clock, a printer output, a second cassette, and a serial card to provide an RS-232-C I/O port (Figure 6-2). Under the

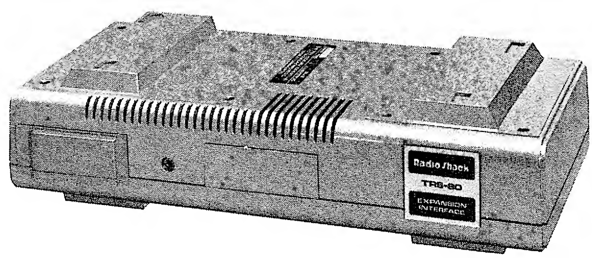


FIGURE 6-2. The Expansion Interface for the Radio Shack TRS-80 Model I holds optional memory, disk controller, clock, and RS-232-C serial interface devices. (Photo courtesy of Radio Shack.)

standard configuration, it is necessary to have the Expansion Interface and serial card to use the TRS-80 as a terminal. Radio Shack originally marketed a separate RS-232-C/Bell 103 modem which was the Novation CAT wearing a Radio Shack label. A more recent modem is an "in-house" development called Modem I which operates in a standard fashion, but which also has a capability to interface to the keyboard computer system without the Expansion Interface (Figure 6-3). A special cable and program is needed, and only half-duplex operation is possible without the Expansion Interface. Radio Shack also has a Vidtex cassette-based program for data communication.

All of these pieces together allow the system to communicate.

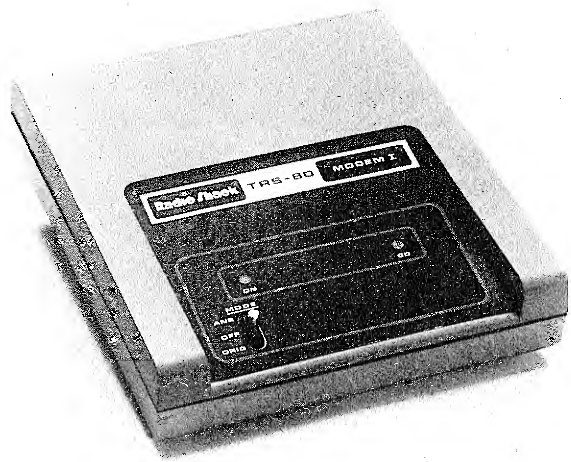


FIGURE 6-3. Modem I from Radio Shack is a complete RS-232-C modem with both direct connection and originate/answer capabilities. With optional software and cable it will interface with the TRS-80 Model I without the use of the Expansion Module. In this optional mode it will only provide half-duplex service. As an RS-232-C modem it will work with any computer or terminal and provide full operating options.

Modem II is a more sophisticated microprocessor controlled device which can accept commands over the RS-232-C lines from any computer or terminal. Its internal programming gives it a complete automatic answer and dial capability. (Photo courtesy of Radio Shack.)

But the Expansion Interface was an expensive option and some individuals who were happy with 16K and cassette storage objected to buying the Expansion Interface simply to be able to add the serial data card. Similarly, the RS-232-C serial data card was an extra cost, and many users wished to integrate a modem device directly into the computer bus. Of course, other users were not happy with the Radio Shack software and wanted a "smarter" software package. The industry responded with a flood of telecommunication hardware, software, and integrated packages.

There are many ways to make the TRS-80 Model I into a data communication terminal. Let's start at the telephone end of the system and work our way back up the bit stream and look at the options as we go along.

The Radio Shack/CAT modem worked well, but as originally marketed it was an acoustically-coupled device. Many users substituted direct-connection devices such as the D-CAT and gained the advantages of direct connection. The more recent Modem I is a direct-connection device. The Radio Shack serial card uses an RS-232-C standard DB-25 connector, so any RS-232-C/Bell 103 modem will work with it.

The Expansion Interface is the biggest target of alternative communication devices. The double cost of the serial card and the Expansion Interface created a situation begging for improvement.

One approach to replacing the Expansion Interface was taken by the Micromint, Inc. They market a device called COMM-80 (Figure 6-4) which serves as an intelligent interface box. COMM-

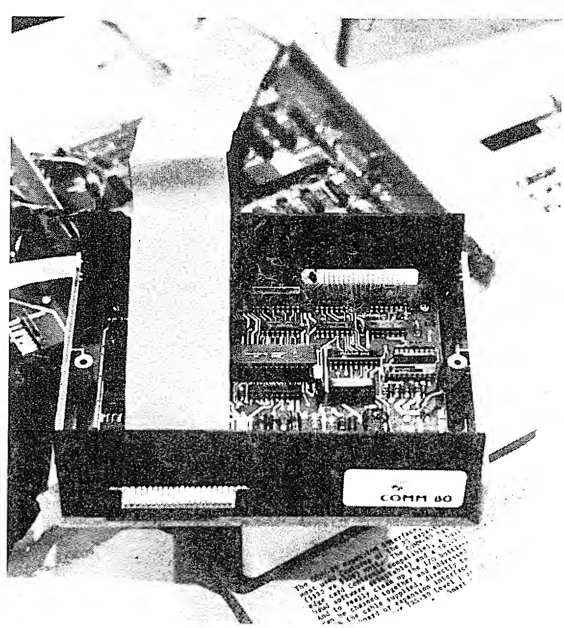


FIGURE 6-4. COMM-80 provides an RS-232-C port for the TRS-80 Model I without the use of the Expansion Interface.

80 comes complete with connectors and cables to mate with the TRS-80 keyboard computer package. It has its own power supply and mates with the computer data bus.

COMM-80 provides an RS-232-C port. This unit is a useful option if you already have a modem or want RS-232-C signalling for some other purpose, but don't want to invest in the expansion interface and serial card. The heart of the unit is a universal synchronous/asychronous receiver/transmitter (USART) which translates between the parallel signals on the TRS-80 data bus and the serial format. The device has switches for selecting data transfer rates and formats. Options on the circuit board control the number of stop bits and other options. COMM-80 can be used to interface with any standard modem or printer using RS-232-C signalling. Up to 16 COMM-80 devices can be attached to one TRS-80. Each device can be given its own unique address code. With the proper software, these devices can provide up to 16 RS-232-C ports for unique multi I/O applications.

THE LYNX

EMTROL Systems, Inc., is an old and established firm in the area of manufacturing controllers and processors for industry. They have entered the microcomputer market in the area they believe has the best potential for growth data communication. They are marketing a modem and serial interface called the LYNX (Figure 6-5).

The LYNX provides a complete direct-connect modem capability for the TRS-80 Model I and III without the need for the use of the Expansion Interface and serial card (serial card only for the Model III).

LYNX uses telephone modular plugs and jacks (type RJ11) to connect directly to the telephone system. It plugs in series between the wall socket and the phone. When the LYNX is in use, the phone is disconnected. If your phone is not equipped with modular jacks, you may have to ask your local telephone company how to make the connection. Many electronic supply stores and telephone stores have adapters to connect the modular jacks to the older-style square 4-pin system. They may be provided at no charge from some telephone companies. Cables with a modular

FIGURE 6-5. The Lynx modem connects directly to the data bus of TRS-80 Model I and III computers. Versions are available which provide complete auto-answer and auto-dial capabilities.

jack at one end and spade lugs on the other are also available. The LYNX is FCC-certified for direct connection into the telephone system.

The other connections on the LYNX are for the power supply and the data bus. LYNX uses the same data ports as the Radio Shack RS-232-C serial card, so the serial card cannot be in the system at the same time. Any standard TRS-80 Model I or III software package will work with the LYNX.

The LYNX is simple to use. A back-panel switch selects between the originate or answer mode. The front panel has one big switch labeled "Talk" and "Data." Two light-emitting diodes show if AC power is on and if a carrier has been detected by the modem's receiver. The parity, number of stop bits, and transmission mode are set by internal switches, but the software can override these settings by keyboard command.

The LYNX also comes with its own software package on cassette, but any program designed to run with the standard TRS-80 data configuration will run with the LYNX. One version of the LYNX EMTERM program is designed for the very elementary and inexpensive TRS-80 Level I system (4K of memory and Level I BASIC). This means that a complete terminal system can be put

together using only the simplest TRS-80 Level I, the LYNX, and the simple version of the EMTERM program. This combination gives good capability at a very low price. LYNX sells in the $280 price range.

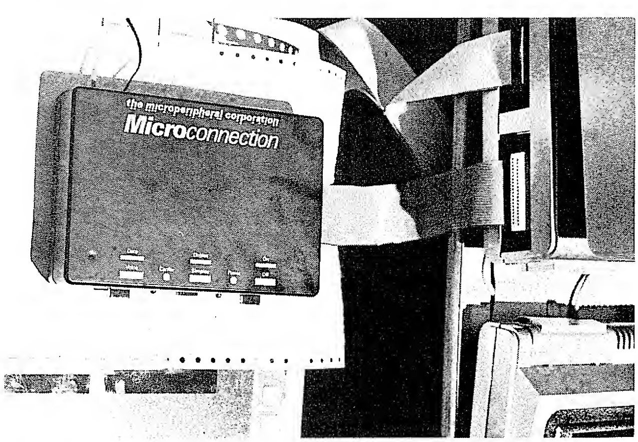
The only major capability missing from such a system is the ability to drive a serial printer to capture on paper the information that marches by on the screen. The next system we will review provides that capability.

THE MICROCONNECTION

The Microconnection is manufactured by the Microperipheral Corporation in Washington State (Figure 6-6). Several different models are available, including an RS-232-C device which functions as a standard modem. The bus-decoding version of the Microconnection can probably best be viewed as a kind of magic "black box." It is a 3-way translator. If you put audio modem tones into it, both RS-232-C signals and parallel 8-bit bus-addressed signals come out. Similarly, RS-232-C signals come out as audio tones and parallel data.

The audio portion of the Microconnection is a direct-connection FCC-certified modem, again using the modular RJ11 plugs.

FIGURE 6-6. The Microconnection can connect to either the keyboard computer or the Expansion Interface. It provides full modem, audio, and RS-232-C interfaces.



The Microconnection plugs into the phone circuit in parallel with the telephone by using a "Y" or duplex modular jack. This is available from the Microperipheral Corporation or Radio Shack (Radio Shack part #279-357). This type of connection allows the phone to be used as a monitor and local testing device. Additionally, the audio lines in and out of the modem are available on separate jacks. This unique feature allows audio recording and transmission of tones from a high-quality recorder and it provides a very handy interface for amateur radio operators who want to use a modem on the VHF ham bands.

The audio recordings of modem tones can be stored and played back later for print-out or transfer to another system. Programs and data can be played back through the modem ports of microcomputer systems having very different cassette or disk systems. High-quality recorders with good speed accuracy must be used for this purpose, but audio recording gives a very practical alternative to disk file systems for storage of material, too. The modem does not have to be connected to the phone line to use the recording or playback feature.

The Microconnection RS-232-C port uses a standard DB-25 female connector. This connector is wired as Data Communications Equipment (DCE). It transmits data out of the Microconnection on pin 3 of the DB-25 and receives it on pin 2. If you want a paper copy of the data received in the modem port, you only have to attach an RS-232-C printer to the DB-25 connector. This same port can be used to drive the printer for more traditional uses without the modem being hooked to the phone lines, but special software addresses for the unique Microconnection port must be used (Figure 6-7).

When used with the TRS-80 Model I, the Microconnection plugs directly into the expansion port on the keyboard computer module or the screen printer port on the expansion interface. The Microconnection also has an auto-answer, auto-dial option.

In operation, the Microconnection offers great simplicity and flexibility. The basic unit has three switches. One switch puts the device into the telephone circuit. The second selects between originate and answer operating modes and the third switch turns the power off and on.

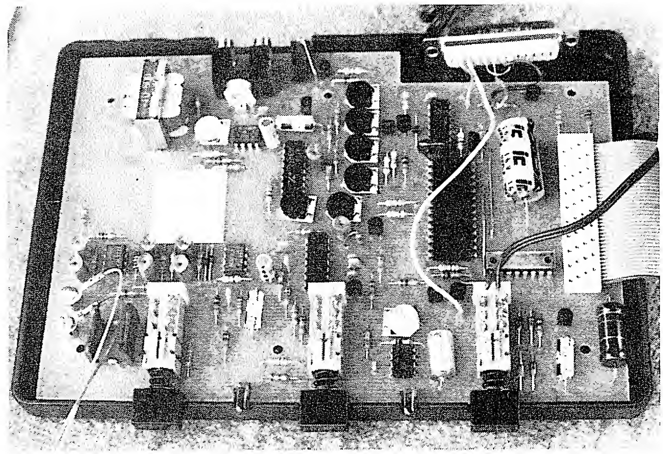


FIGURE 6-7. This interior view of the Microconnection shows the DB-25 connector and audio jacks at the top, various function switches at the bottom, and the parallel interface cable on the right. The telephone line is attached to the screw posts at the left.

The instruction manual for the Microconnection is superb. It describes methods of using the telephone for local tests, special telephone situations, audio recording, amateur radio operation, troubleshooting, and much more. The instructions are detailed, yet very easy to follow. The instruction manual also provides details on the S80 terminal program supplied with each unit.

The S80 terminal program provides simple terminal operation. It comes on cassette and will work with any TRS-80 Model I, even the Level I 4K system. The bus-decoding version of the Microconnection uses a port address which is different from the standard TRS-80 configuration. This means that software for this system must use port 208 for data and 209 for status messages and meet the Microconnection protocol. Several dealers have software packages optionally configured for the Microconnection, but standard software will not work with this device. An advantage of this addressing is that a sophisticated user can retain the original addressing with an Expansion Interface (or LYNX or COMM-80), use the Microconnection at the new address, and have two independent or interconnected serial ports.

The Microconnection line of communication devices is

available with many options. The auto-answer provision permits unattended remote access of the host computer. A detector is provided for ring counting and preset answer conditions. Auto-dial comes with the auto-answer version. Another option allows the Microconnection to be used with European standard modem tones. Versions of the Microconnection for TRS-80, ATARI, and Apple computers sell for about $250; various options are extra. An RS-232-C only version (actually a standard modem) sells for under $200.

So far we have described four different hardware combinations that will provide a communication capability for the TRS-80 Model I. The various systems vary in price and capability. Each user can have the fun of designing an individual system based on available equipment and needed capabilities. A major part of every data communication system is the software, and there is a great deal of communication software on the market.

TRS-80 MODEL I AND III SOFTWARE

Many different kinds of software are available to make the TRS-80 Model I and III into a communication terminal. The programs vary in the complexity of the hardware they require to operate, the degree of "intelligence" they give the system, and in price. One fact all software designers encounter is that the Model I has a limited keyboard and ASCII-character generation capability. The TRS-80 Model I does not come equipped to deal with control codes or lower-case characters easily. Most information utilities and message systems operate more efficiently if control codes are used. Various software designers have met the keyboard limitation in different ways. Some simply ignore control codes and leave the user to work around situations where their use would speed service. Others have redefined or redesignated certain keys to produce control codes under certain conditions. Each of the programs available provides its own combination of requirements, flexibility, and cost.

The EMTERM program that comes with the LYNX is a good program to examine first. It loads in from a cassette and can be transferred to disk. It does not have an extensive file-handling capability. It provides for the transmission of control characters and has several nice features. It can be described as economical and practical. It comes with the LYNX at no added cost and will work with the "bare bones" TRS-80 Model I or III.

EMTERM, like most communications programs, is menu driven. A menu of alternatives is presented on the screen and the desired alternative is chosen by entering a number or letter. The main menu provides the alternatives:

STORE MESSAGE	(S)	
RECEIVE BASIC	(R)	(Level II systems only)
TRANSMIT BASIC	(X)	(Level II systems only)
TERMINAL	(T)	
VIEW/CHANGE UART	(V)	
BASIC	(B)	

The "STORE MESSAGE" feature allows preparation and storage of a message off-line for later transmission. Message storage is done in a 1K buffer allocated by the program. This feature allows the user to save valuable on-line time because the message goes out much faster than the average individual can type.

The "RECEIVE BASIC" option for the Level II system allows the receipt of a program written in BASIC from the distant system. EMTERM contains a special relocation feature that tucks the communication program away in high memory and leaves the normal RAM workspace free. The received program is saved in the RAM workspace and then can be RUN, LISTed, or SAVEd. This program transfer is best accomplished with another system running EMTERM and using the TRANSMIT BASIC mode.

The "TERMINAL" option provides the normal conversational mode of operation. In this mode, all data received is lost

when it scrolls off the screen and all data sent must be entered on the keyboard. It is possible to toggle a printer connected to the parallel port of the TRS-80 on and off to allow simultaneous printing of the displayed characters.

Some system diagnostics are presented by this program. If a mismatch of either parity or word length (the framing bits) occurs, diagnostic letters appear on the screen. This is a unique program feature, but often the framing errors detected are not serious enough to interfere with character transmission.

The parity, word length, and number of stop bits can all be decided from the keyboard. These entries temporarily override the switch settings on the LYNX circuit board.

The EMTERM program provides a message buffer and BASIC program transfer capability. It also can provide for the transmission of control codes including "BREAK" and "ESCAPE." An output buffer also is provided to accommodate slow parallel printers.

Program start-up is simple. After setting "memory size" and loading EMTERM, the command "SYSTEM," ENTER, "/" ENTER brings the question "New starting address?" up on the screen. Following guidelines in the instruction manual, answering with a 4-digit hex-relocation address sends EMTERM off to high protected RAM or any chosen block of memory. After EMTERM is stored in protected RAM, BASIC programs can be loaded and run normally (except that a small portion of memory is not available). The BASIC command "SYSTEM" ENTER, "/" (relocation address in decimal) will restore the EMTERM menu without affecting the BASIC program.

In practice, EMTERM is a practical program for use with various information utilities such as the Source and CompuServe. The S80 program provided by the Microperipheral Corporation with the Microconnection provides most of the same capabilities.

S80

S80 is also a cassette-based program designed to run with the simplest hardware. The S80 program uses the up arrow to shift into the control code mode. The number keys can also be rede-

fined to provide some of the brackets and other symbols sometimes used by large systems, but that are not available on the TRS-80 Model I keyboard. The action of a printer can be controlled through the parallel port. There is no off-line message preparation or simple program-exchange capability.

While these two programs come "free" with equipment, they are fairly representative of about ten elementary communication programs available for the TRS-80 Model I. They provide a good basic communication ability, but each has its own trade-offs and limitations. Data communication users quickly find they want more powerful programs and there are several good ones available.

SMART PROGRAMS

There are certain functions and attributes a good smart communication program should have. The list of absolute "must have" items includes:

- Excellent documentation including all operating information and details on customizing programs. A good program must have a good operating manual.
- The ability to prepare files off-line for later transmission. If messages and programs can be read in and automatically transmitted, a great deal of money can be saved in long-distance telephone calls and system use.
- Buffer storage that can be turned off and on. The ability to quickly open and close the buffer allows a kind of rough pre-editing of the data to be saved. It saves time and paper during later printing.
- Prompted transmission of lines or characters from the buffer. Some systems prompt for transmitted data. If a communication program will not wait for and recognize the prompt, a great deal of data can be lost.
- The ability to transmit frequently needed strings of characters (macros) with the touch of a key. These macros greatly simplify signing-on and using systems.
- Complete support of all control characters and hopefully lower-case characters, too.
- Ability to transfer files stored in a format other than ASCII. (These files are converted to ASCII before or during transmission.)

- Some degree of screen formatting (for example, words do not break in the middle if the transmitted line is linger than the display screen).
- Control over an auxiliary printer.

There are several programs which come close to meeting this list, but at least three match it perfectly and add some frills of their own.

SMART 80

The SMART 80 series of programs is written by Dick Balcom and is marketed by the Microperipheral Corporation which produces the Microconnection. Several different versions of the program are available to support different equipment combinations.

Since one version of the Microconnection is designed to be used without the Expansion Interface (which holds the disk controller) a SMART 80 program is available on cassette. It loads files to and from cassette almost as easily as other programs do with a disk. It is the most powerful cassette-based communication program available for any microcomputer system.

Another SMART 80 version works with Stringy Floppy high-speed tape system manufactured by Exatron. This combination gives a very nice low-cost alternative to disk-drive storage. The only thing a disk-drive system can do better is to make frequent entries into a data file. Frequent file access is very uncommon in terminal operation.

Of course, a very powerful disk version of SMART 80 is available which is compatible with all major TRS-80 Model I disk-operating systems, including double density.

SMART 80 adds some features to the above list of "must have" items. These include an automatic upload and download of files from CompuServe and Forum 80 systems and between users, transmission of assembly language files, and a speed-selection feature that will slow the throughput of characters to insure that the receiving system receives all the data, even if it is heavily loaded and running slowly. Two prestored messages (macros) of up to 30 characters are available. The keyboard is provided with an automatic repeat key function and a "beep" tone is sent out of the cassette port each time a key is depressed. A speaker connected to the cassette port will provide positive feedback on each keystroke. This greatly improves the "feel" of the keyboard sys-

stem—especially for typists used to mechanical systems where something goes bang each time a key is depressed.

The SMART 80 software carries a price of between \$80 and \$100 in the various versions. It is available in versions addressing either the Microconnection or the standard TRS-80 operating ports.

ST-80

Another available series of programs is called ST-80 and is written by Lance Micklus. The series includes low-cost communication software, but it is capped off by a fine program called ST80-III. ST80-III also fits the definition of smart terminal software given above. It provides for one prestored sign-on message of up to 63 characters. A unique feature of this program is the option for encrypted transmission. Files can be moved into RAM, converted to ASCII if they are not already, and scrambled by a random number generator driven by a 25-character password. They can then be saved again for later transmission. The scrambled files can only be unlocked by someone using the ST80-III program and having knowledge of the password. This feature may be particularly useful where industrial security or personal privacy are important.

ST 80-III features a set of software translation tables which work on all transmitted and received data. The translation tables allow detailed customizing of the software so it can communicate with practically any system including large mainframes. Some large-system manufacturers make unique use of control codes during data transmission. ST80-III can be instructed to translate codes so they have meaning to both systems or so the TRS-80 ignores the codes it does not need. As an example, some mainframe systems send control codes for data formatting which are used by modern printers for controlling the type size. A printer copying along with the data input might suddenly start reacting very strangely if these commands are passed along exactly as received. The ST80-III translation table allows a tremendous degree of flexibility and sophistication to be programmed into the TRS-80 Model I.

ST80-III includes a feature called Veriprompt™ which checks echoed characters against transmitted ones. This feature insures accurate transmission of data. The characters are checked "on the fly." The system does not send one character and wait for

the echo, but rather it looks back to compare transmitted characters with echoed ones while still sending new characters. At most, two characters are in the system at a time.

ST80-III has many other operating features including the ability to execute any of the disk-operating system commands directly from the program. This allows great flexibility in operating and file manipulation.

This software for the Model I sells for about $150 on disk.

OMNITERM

OMNITERM is a program written and distributed by David Lindbergh. OMNITERM also uses a translation table system, but it is more extensive than the table system found in other software. A group of seven translation tables check every character going to and from the keyboard, to and from the disk-operating system, from the keyboard, and out to the printer and the screen. It is especially designed to use both ASCII and other codes such as EBCDIC with equal ease (but not Baudot; it can't handle the shift code). OMNITERM has other unique features such as the ability to see text that has scrolled off the top of the screen and the ability to fully reformat the screen so that 80, 40, 32, or any other column width will appear neatly as 64-column text on the TRS-80 screen.

The features of OMNITERM will please the most sophisticated users. It has many special transmission commands which make it particularly useful for persons who check into several different kinds of systems. Any number of special customizing files can be created which can be ready by the program and used to modify the program to suit the system to be communicated with and the peripheral devices in use.

The screen-formatting capability and on-line scrolling display of the data buffer greatly aid in the use of electronic mail systems. Many message systems send 32-column lines because of the display-size limitations found in some microcomputer systems operating as terminals. A TRS-80 Model I, II, or III does not have this display-size limitation. OMNITERM will change the carriage returns in these short lines to a space. It will fill a line as full as possible and insert its own carriage return when the next space character is found. This prevents the fragmented lines found in

most "wraparound" programs and provides the user with full, clean lines of text.

Screen formatting makes messages look better and easier to read and on-line buffer access allows quick recall of message specifics. This is particularly useful when writing an on-line reply to an electronic mail message. If you can quickly look back to the message you just received, you can pick up details important to the reply you are composing.

This look-back feature gets around one of the great disadvantages of electronic mail (as it is presently formatted) over paper mail. Electronic mail usually requires you to compose a reply without direct reference to the note you are replying to. A written piece of paper is usually available for reference when replying in the paper mail system. The OMNITERM program allows the user to scroll back through the buffer to see the received message again so specific points in the message can be commented on. OMNITERM is a sophisticated program, but it is easy to use because it is menu-driven. The program comes with a 60-page instruction manual that is outstanding. The manual includes an index and glossary and is unique in the industry. OMNITERM sells for about $100 on disk.

UNITERM/80

Apparat, Inc., of Denver, Colorado, has released a very flexible program which can adjust itself to meet the various communications configurations a TRS-80 Model I or III might be in. Uniterm/80 is a complete smart terminal program with all of the standard features such as file transmission and reception and smooth integration with a printer. It is very flexible and provides complete control over the transmission parameters and file activities. Translation tables are provided for the popular information utilities and electronic message systems. It also has special features such as character checking during file transmission and the ability to modify the capture buffer before it is saved to disk. It allows for the creation of overlay files which customize its operation (preloading sign-on macro messages and so on) for specific message systems or information utilities. It also has a unique ability to ex-

amine the serial port addresses of the computer when it initializes and automatically configures itself for proper operation with a standard serial card, Lynx, or Microconnection.

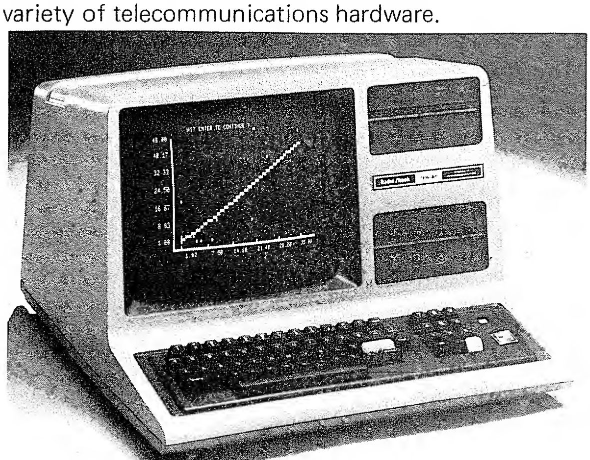
Uniterm/80 will first determine if a Microconnection is connected to the address bus at port 208. If it is not, then the program will look for other active devices. This is very useful for devices such as the COMM-80, which has variable addressing, and the Lynx, which can be used at many different addresses on the TRS-80 Model III.

Uniterm/80 is designed to be used with Apparat's special disk operating system for the TRS-80, NEWDOS/80. The program will, however, operate with almost all features under the standard TRS DOS. Uniterm/80 costs about $80 and is available from Apparat, 4401 S. Tamarac Parkway, Denver, Colorado 80237.

Hardware and Utilities

Most of these intelligent terminal programs require at least 32K of RAM to operate. (Some SMART 80 programs can run in 16K, but you will have a small buffer.) They can operate with the various modem systems described in this chapter; all but Uniterm/80 require special versions for operation with the Microconnection,

FIGURE 6-8. The TRS-80 Model III is a fully integrated computer system. Most smart terminal software available for the Model I now comes in versions for the Model III. It can operate with a variety of telecommunications hardware.

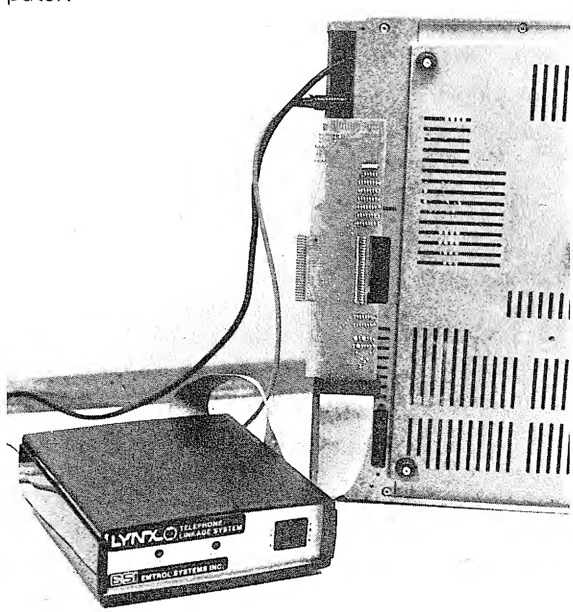


or other devices used at different port addresses. They all come with some utility programs performing various functions. The most common utility is one allowing the creation and saving of off-line messages. Other utilities may change files to ASCII or compress files for more economical transmission. A TRS-80 using any of these programs will become one of the most powerful data communication devices available today.

THE MODEL II, III, AND COLOR SYSTEMS

All of the large computers in the Radio Shack line have some capability to communicate. The Color Computers come with an RS-232-C port as a standard part of the equipment. The Model III

FIGURE 6-9. The LYNX modem connects to the TRS-80 Model III in a unique manner. The special adapter circuit board plugs into the data bus-expansion port and extends it out so it is still available for other accessories. In this way, the LYNX can operate without a serial port while still not occupying the data bus port. The dual ended card allows the LYNX to sit on the right or left side of the computer.



(Figure 6-8) needs only an RS-232-C card as an extra cost hardware option (it is provided with the dual-disk 32K package). The Model II comes with two built-in RS-232-C ports. The systems need a standard modem to interconnect with the telephone line.

Unique communication software for the Model II will be discussed in Chapter 7. The ST80-III terminal program is available for the TRS-80 Model II.

The Color Computer has a special Radio Shack Videotext program providing a terminal capability specially configured for the CompuServe information utility. The ability to locally prepare messages off-line for later efficient on-line transmission is provided. The program allows the transmission of control codes, and it recognizes the cursor positioning and screen formatting commands (for example, clear screen) sent by CompuServe. Upper- and lower-case characters can be transmitted. The Color Computer running the Videotex communication software can be used to communicate with any message or information utility. The program on a cassette plus one hour of CompuServe computer time are available from Radio Shack stores for about $30.

Almost all software packages on the market today are compatible with either the TRS-80 Model I or III. The only major difference is the disk format. The common practice is to ship disk based programs in Model I format and use the CONVERT utility to change them to Model III format. A Radio Shack Videotex communication package, similar to the one described above for the Color Computer (but without off-line message preparation) is available for the Model III on cassette. The program will load and run on either the Model I or Model III and is very easy to transfer from cassette to disk.

None Better at Any Price

A TRS-80 microcomputer running one of the smart terminal programs can become one of the most sophisticated data communication terminals available at any price. The broad capabilities built into this line of computers insures that any level and degree of sophistication needed is available.

7

The TRS-80 Model II

Each machine in the TRS-80 family has its own personality. The Model I was an explorer. At first it was slightly eccentric, but it settled down to a productive maturity. The Model III has built on the accomplishments of its older brother, but it probably won't be the pioneer that the Model I was. The Color Computer seems to be more of a domestic device. It can deliver great power to the home, but it is only moderately powerful itself. The Model II, however, has led a different life. It quickly went off to answer the call of the corporate office. It uses different disks and different programs from all of the other TRS-80 systems. The wider display screen and complete keyboard provide more capability for entering and displaying business programs, and the large disk provides good storage in a format that can be carried to other common small business computers.

The TRS-80 Model II comes equipped to communicate. Its two RS-232-C ports can be used to feed a modem, printer, and other serial devices. A very capable terminal utility program comes on the TRSDOS disk at no extra charge. The manual details the operation of the hardware and software, but there are two areas where users might have a problem.

The two RS-232-C ports can each be turned off and on from the keyboard, but if only one port is in use, the other must have a proper electrical termination. Termination is done by inserting a male DB-25 plug with pins 3,5,6,7, and 8 connected together into the unused port. This plug does not come with the computer, but it is easy to make if the user has an elementary skill with a soldering iron. Failure to use a termination plug may result in improper operation of the active port.

The TERMINAL program on the TRSDOS disk provides a full line of capabilities to act as a terminal, save received data in disk files, transmit data from files loaded into RAM, and perform other practical functions. It operates from a displayed menu and is easy to use in most ways, but an inexperienced user might be dismayed by the way the program displays transmission errors.

The digital bits in an asynchronous ASCII stream often get jumbled. Noise pulses on the line, quirks in electronic timing circuits, and slightly different operating parameters all can combine to create errors in parity or synchronization. Only a tiny fraction of these errors ever result in one lost or garbled character. When the TERMINAL program detects parity of framing errors, it displays a reversed video flag in the text. These flags can create a great deal of confusion if you don't know what they are. If you receive a great number of flags, the system you are operating with may be using different transmission parameters. You will have to either find out what those parameters are or experiment using the SETCOM function of TRSDOS to change your settings until the errors become less frequent. It is important to remember that the serial port must be turned on with a SETCOM command when using TERMINAL and that it must be turned off before the parameters can be changed. Understanding the error displays and the use of the SETCOM function before you begin can save hours of frustration when you are on-line with the Model II.

BiSync COMMUNICATIONS

The TRS-80 Model II is one of the few microcomputers with software allowing it to communicate directly with IBM mainframe systems using the IBM BiSync (binary synchronous) communication protocols. IBM has used these communication protocols for many years, and a great deal of IBM equipment has no asynchronous communication capability. Radio Shack markets two BiSync communication packages: one for the IBM 3270 protocol and the other for the IBM 3780. Each package costs about $1000; but if you are a commercial user of the TRS-80 Model II with a need to communicate with an IBM mainframe, this price can be quite reasonable.

THE MODEL II AND CP/M

The need for "portable programs" in the business world has led to the popularity of a de facto standard operating system for 8080/Z-80 systems called CP/M. CP/M is an operating system designed by Digital Research and "CP/M" is their registered trademark. Just as the S-100/IEEE 696 bus has become a standard for hardware connection, CP/M has become the software interface standard or the software "bus" (though not an IEEE standard). CP/M is not unique to any particular computer bus system. Many microcomputers built on a single board or with other interconnect structures can and do operate under CP/M including the TRS-80s, Heath, and even the Apple II with a special additional Z-80 CPU.

CP/M is an operating system like TRSDOS. The operating system is the most fundamental level of software. It is always online during normal operation, and most of the time it is "transparent" or at least "opaque" to the user. It is the first level of software to talk to the user. When you first tell your computer to load a language (BASIC), you are probably talking to the operating system.

The exact line drawn around what is the operating system may differ between different kinds of equipment. Most microcomputers have some sort of monitor program (usually in ROM) looking for keyboard inputs and other signals. Many disk systems

load a disk-operating system in from the disk which integrates with the monitor and forms the operating system. Operating systems may be contained totally in ROM, or they may be read in totally from tape or disk after the system is "booted" alive with a wired or switched-in start-up routine. An operating system provides the software routines to move data in and out, put it on the disk, and interface with higher-level languages.

In theory, if everyone used the CP/M operating system, identical higher-level languages and programs could be loaded into all machines. This theory doesn't hold strictly true because some software is written to work only with computer systems using a separate terminal or some other software combination. Some computers capable of operating under CP/M (such as the Model I and III) have monitor ROMs residing in low memory where almost all CP/M programs are written to run. CP/M programs for these machines have to be rewritten or somehow "moved" so they operate in higher memory space.

Use of the CP/M operating system also does not infer total disk portability. There are factors such as physical disk size (5-1/4- versus 8-inch) and format (density) which limit the ability to carry a disk from one CP/M machine to another. The best interchangeability comes between so-called "big disk" (8-inch) single-sided single density systems such as the one used on the Model II. A CP/M disk made in this format can be carried to most other 8-inch CP/M systems, and the files on the disk can be read with no problem. This is not true of most 5-1/4-inch disk systems running over CP/M; these small disk formats remain unique. The same programs can be run on all systems if they can be entered in (and, in some cases, properly address in memory away from permanent ROMs), but getting the programs into it may still require a disk format unique to the specific disk hardware in use.

Once the CP/M operating system is customized to the unique hardware it is on, the details of the hardware (aside from the disk differences) become almost irrelevant. (Note the "almost.") It presents a "universal" face to the outside world while internally doing the unique routines needed to run the specific machine. A program usually need only be designed to run with CP/M, not with any particular microcomputer or disk system.

Several microcomputer manufacturers have developed operating systems which use the CP/M standards for software interface.

Cromemco, for instance, has a disk-operating system called CDOS which is not CP/M, but it will respond to CP/M commands and interface with CP/M programs.

The CP/M package consists of a monitor control program plus some utility programs. The disk includes utility programs such as a text editor, an assembler, and a debugger. This total package enables the user to create, edit, debug, assemble, and run programs. It can be used with any of the 8080/Z-80 microcomputers running with at least 16K of RAM. Versions of CP/M for almost all 8080/Z-80 systems are available from Lifeboat Associates. The prices run between $150 and $400 for a complete package on disk. The Model II package is under $200. A very nice CP/M package customized for the TRS-80 Model II is also available from Pickles & Trout. These customized packages are essentially ready to run on the Model II and do not require the knowledge of programming needed to install CP/M on some other systems.

Another very positive benefit of using CP/M comes from the CP/M Users Group. This organization maintains a huge library of public domain (that means free) software. Lifeboat Associates will provide copies of the Users Group programs for a nominal disk and copying fee. Local chapters of the CP/M Users Group can be found in many cities. A newsletter, LIFELINES, is available for a yearly subscription fee. The newsletter covers the activities of local groups and product information. The address for subscriptions can be found in the Appendix.

One caution is probably in order. CP/M is a complex system that requires some sophistication on the part of the user. The documentation has been criticized by many. Before making a big financial investment, you should seek out a local person, users' group, or computer store with experience in the CP/M and the hardware you will be using.

LET'S COMMUNICATE!

CP/M provides us with the capability to use a whole world of programs on different 8080/Z-80 machines with no modification. The Users Group can provide disks in many different formats, but

that doesn't help individuals who may want to exchange their own programs between computers. Disks are not completely portable between machines, so how can we get the programs inside our microcomputers without the tedious work of typing them in from the keyboard? As you are this far in the book, you know that a data communication capability is the obvious answer. Before we can communicate, we have to add the right pieces to our Model II CP/M puzzle. Those pieces are a modem and software.

The Model II comes with two serial ports. One is typically used for a modem and the other for a printer. Any standard RS-232-C modem will work with the Model II.

The Model II needs some kind of software to operate as a terminal. The TRSDOS Teminal Utility is a capable program, but the files it writes under TRSDOS cannot be read by CP/M business programs. (However, some utility programs to convert between Model II TRSDOS files and CP/M files are available.) The world of CP/M software is confusing. You can pay a lot of money for some commercially marketed software or very little money for a "public domain" program that might serve your purposes. Obviously, some detailed research is needed to insure that either kind of program actually meets your needs. The cheap program that does not do all that you want is no bargain.

ALMOST FREE?

The CP/M Users Group has several pieces of public domain communication software available through local chapters or the national headquarters which shares a mailing address with Lifeboat Associates. Some fees are involved, but these are basically for membership, service, or disks.

MODEM is certainly the most widely used CP/M communication program. It was written by Ward Christensen who, as we will discuss later, is one of the founders of the Computer Bulletin Board System. MODEM is a straightforward program allowing conversational-type terminal operation. It also has a "block mode" transmission feature that will very accurately transfer files with another system running under MODEM. Checks and acknowledg-

ments are sent from both ends, and the system will try to send a block of data many times over before it quits.

PLINK is both a conversational and file transfer program, but it simply dumps the file out the serial port. It doesn't do any checking with the distant-end system to verify correct receipt. This program is often known as a "de-spooler" because it just runs the data out like tape or string off a spool. This is a handy mode of operation and very adequate for many uses such as electronic mail. It can be troublesome with some systems which insist on digesting data by the character or line and then calling or "prompting" for the next input.

MODEM and PLINK are available from the CP/M Users Group for a nominal fee.

CROSSTALK

CROSSTALK is an intelligent communication program for CP/M disk-operating systems marketed by the Microstuf Company. It costs about $150 and provides a very sophisticated set of capabilities.

CROSSTALK uses specially-named configuration files to store the important data about each system you communicate with. The telephone number, baud rate, parity and stop bits, and transmission mode (half- and full-duplex) are all stored in files dedicated to each system. When you load a file, it customizes the program to the system you want to communicate with. It also supports the use of the Hayes Smartmodem for auto-dial and auto-answer communication.

The program contains a "debug" mode which will display on the screen the hexidecimal value of each character being received in the modem port. This is very handy when you are trying to figure out exactly what the system at the other end is sending you in the way of control, end of line, or possibly special graphics characters. This is a great aid in customizing the communication system for the most efficient transmission.

CROSSTALK has an easy-to-use protocol file transfer system that can send files directly from a disk file on one system to a disk

file on another CROSSTALK-equipped system. This transmission does not pass through a buffer, so it is not limited by the size of the memory available on either system. The program includes elaborate error detection, automatic retransmission, and timeout recovery.

In addition to the protocol file transfer, CROSSTALK will transmit files to almost any other electronic mail or data system using a program which compares the echoed character to the transmitted one. The program will input data into systems such as an ABBS which requires line-by-line prompted input, but it does so using a time delay instead of an actual prompt-response action. Other features include an ability to transmit files from any word processing system, a graph which shows the amount of buffer space filled with text, and a complete menu-driven command table. CROSSTALK is a very capable terminal program for Model II CP/M users. It comes in a ready-to-run customized version for the TRS-80 Model II.

The Grey Flannel Computer

The TRS-80 Model II is at home with the professional programmer and businessperson. The CP/M operating system provides another way for the system to interact with other large-disk computer systems.

8

Microcomputer-Based Message Systems

The early chapters of this book introduced the concept of the Time Tyranny of Telecommunications. Defeating this time tyranny requires some form of store-and-forward message system that will hold data messages until the addressee desires delivery. While some different kinds of store-and-forward systems (including telephone-answering machines) are now popular, they do not provide all the advantages of a data communication electronic message system. Any really useful message system makes entry and retrieval of messages as simple as possible. The ability to transfer something more than just messages is also desirable.

A store-and-forward message system has many practical applications in commerce, industry, education, and training. Travelling sales personnel can dial into such a system at the end of a day's work and pick up calls, leave orders, and skim the latest market and sales information. This can be done without detaining

or employing people at the main office to answer telephones and take messages. Special programs can be available on the system to help figure orders, costs, or commissions. Excellent microcomputer systems that can do this cost less than $5000.

Industry can use a store-and-forward message system for any group of individuals in different locations working on similar projects. Development teams and project teams can trade information, receive assignments, and keep track of progress using a dial-up message system. Software teams can exchange complete listings or actually demonstrate running programs on the same system.

A message system is an outstanding educational tool for every level from high school through graduate school. High-school students can achieve a degree of computer literacy through interaction with a computer system shared by a class. College and graduate students can exchange class notes, consult with faculty advisors, and run special programs. Evening and correspondence students can benefit from having a class forum available whenever they need it.

All these applications (and many more) can be met with a microcomputer-based electronic message system. Several hundred electronic message systems are in operation throughout the world, mainly in the United States and Canada, but systems have been reported operational in England, Holland, France, Korea, Australia, and Japan. These systems vary in complexity and degree of service. Some run on sophisticated minicomputers with many megabytes of disk storage space available. Others use simple single-board microcomputers and programs which store messages only in the available RAM. Some are able to exchange programs with users, provide protected mail service, and even provide elegant subsystems for the use of special interest groups. Some systems consider themselves national in scope; others serve a few friends living in a local neighborhood. Microcomputer-based message systems are used by large corporations, small shops, private special-interest groups, and clubs of all types.

THE SYSTEM OPERATOR

Someone has got to take care of the darn thing! Although electronic message systems run unattended, they still require regular "care and feeding." After a system is operational, the sys-

tem operator (SYSOP) will probably spend a minimum of thirty minutes a day killing old messages, responding to new ones, and compacting message files.

Most SYSOPs supply everything free (the computer, power, phone line, and disks) because they enjoy serving or because they like being a part of the action.

The average message system in a metropolitan area has about thirty to forty callers a day. Some have many more. Many callers are young people using terminals at local schools. Others have varying levels of expertise in microcomputer systems. Thanks to the efforts of the SYSOPs, the callers can share knowledge and ideas that transcend the usual cultural and social barriers.

This chapter will describe the history, use, and set-up of the most common types of microcomputer-based message systems. First, let us see where it all started.

CBBS #1

Two men living in the Chicago area, Ward Christensen and Randy Suess, were early users of microcomputer systems. In 1977, they began communicating between their systems to swap software and messages. During the winter of 1978, Ward wrote the software for a kind of electronic bulletin board which would replace the actual cork bulletin board used at a local computer club. This system was to work exactly like a cork and thumbtack bulletin board. People entering the computer through the telephone lines could browse through the notes and see what equipment might be for sale, who needed help, and so on. They could leave notes in reply and destroy their own notes after they had lost their usefulness. Randy put together the microcomputer hardware and the first Computer Bulletin Board System (CBBS) was born.

At the time, electronic mail systems were available on several large mainframe computer systems; but Ward and Randy, in effect, reinvented the wheel for microcomputer-based systems. By the middle of 1978, four CBBS computers were in operation. The number has grown to over a dozen, but there is a frequent change-over of systems as individuals gain or lose time, equipment, or interest.

ABBS

In 1979, the Apple computer grew rapidly in popularity. Two men in California shared an idea for setting up an electronic message system based on Apple hardware. Craig Vaughan seems to have developed the original approach. Bill Blue did the operational and human engineering of the software, and the product was marketed through a California firm and computer stores. Thus began the Apple Bulletin Board System (ABBS). Because of the availability of the software and the popularity of the hardware, there are usually more ABBS computers in operation at any given time than any other kind of system. These are often hobby systems run by individuals, and their operation may not be as consistent or as permanent as other systems.

FORUM 80

With the popularity of the TRS-80 computer (particularly the version now known as the TRS-80 Model I), it was inevitable that good message-system software would appear for that hardware. Bill Abney in Kansas City, Missouri, wrote a program he calls Forum 80 for this system. Abney has kept close control over the Forum 80 operation, and the various operators around the country cooperate closely in updating software and maintaining standard methods of operation for the users.

OTHER SYSTEMS

Many other kinds of microcomputer-based message systems now exist. The availability of less expensive auto-answer modems has made interface to the telephone system reasonably easy. The protocols have been initially established and the method of operation refined. Now that the ground has been broken, almost any decent programmer can develop a message program in BASIC that will perform elementary message filing and display functions. However, the first developers deserve a great deal of credit for

taking the lead and providing the software and organization that allowed hundreds of thousands of terminal users to enter the world of personal data communication for the first time.

USING A MESSAGE SYSTEM

Initial success in using an electronic message system depends 10% on technical skill, 20% on terminal equipment, and 70% on psychology. Many technically-skilled persons freeze when a computer asks them their name. Some of the best computer-communication users are liberal arts majors in college and nontechnical students in high schools. Later, experience replaces psychology, but technical skill is never a large factor in the use of electronic message systems.

Many persons, it seems, are either afraid of causing damage to the distant end computer system by giving it incorrect commands, or of looking foolish to the machine. Neither fear is justified. The operating programs for all successful message systems are "bullet proof." They have been refined to the point where they cannot be broken, even by deliberate malicious action. In fact, they are loaded with help commands and useful prompts to direct inexperienced users. Inexperienced users should not be afraid of embarrassment. It is true that SYSOPs often watch users operate the systems (referred to as the "Big Brother position"), but this usually is done with an eye toward monitoring the hardware operation and improving the functioning of the software. If you ever do manage to show the SYSOP a problem, you will probably be met with gratitude. So, when trying a system for the first time or when introducing others to the world of data communication, remember: You cannot break the computer, and it will not laugh at you! It is indestructible, and it has nearly infinite patience.

Sign-on

Now that we have developed the proper psychological attitude, let us see how we go about entering and using an electronic message system. The systems are not standardized, so this description will be given in general terms with some explanation of the exceptions.

The first step is to properly set up your terminal or your microcomputer serving as a terminal. You should be operating as a terminal in the full-duplex (echo-plex) mode with upper-case-only characters selected. Some message systems and information utilities will not accept commands in lower-case characters. Your system should be set to 300 baud, 7-bit words, even parity, and 1 stop bit. TRS-80 users may want to try 8-bit words, no parity, and 1 stop bit. This setting is sometimes used by the Forum 80 systems to send TRS-80 graphics. Your modem should be in the originate mode with full duplex selected. Now you are ready to dial the telephone.

Some terminal hardware/software combinations will dial the telephone for you on command, but let's assume we are using a simple system with an acoustic coupler. As the number gets ready to ring, you should listen to the quality of the telephone line. If there is a great deal of noise, you might consider hanging up and redialing in an attempt to get a better circuit. Remember, your modem will have to detect changes in tones that happen in about three milliseconds. If the telephone line has a lot of noise, those changes could be masked. If the number is not busy, the system should answer within three rings. Four or five rings without an answer indicates the system is not operational.

The system should answer with a high-pitched tone. This is the constant tone coming from a modem in the answer mode. Most systems will only wait about eight to ten seconds without receiving a tone in return before assuming your call is a wrong number and disconnecting. Your modem will take a second or so to recognize the tone and reply, so do not delay too long in placing the telephone instrument in the modem cups or cutting the direct-connect modem into the circuit. If you are using an acoustic modem, make sure you put the handset in the right way! An arrow will often point toward the end of the modem the handset's cord should be on. Also, an acoustic modem must be used in a reasonably quiet area. A radio or a noisy printer can inject unacceptable noise into the handset and modem. Once connected, your modem should respond to the tone and light its READY or CARRIER light or give some other indication it is working.

The first words you see on your screen will probably be some variation of HIT CR. The message system needs to see a few of your carriage returns (the ENTER or RETURN key) so it can tell

what speed you are using. Some systems still support 110 baud, and several support speeds up to 600 baud; the system needs to know how fast you are going so it can set itself to match your speed.

The next question may be: TERMINAL NEED NULLS? This question is important if you are using a printing terminal. Many printing terminals need nulls or blanks after a carriage return and line feed to allow the print head enough time to return to the left side of the page. If this is not a problem, you can probably ignore this question and it will go away. If it does go away, simply hit RETURN. If you are using a slow-printing terminal, follow the directions (type YES or enter control N), and the system will enter nulls. If you are asked how many nulls, try answering 5. You may have to experiment with this number until you get good print head return with little idle time.

Forum 80 systems may also ask if you want line feeds. Line feeds move the paper up in a printer (or the cursor down on a screen) and are not always associated with carriage returns. This is a function of your software. Some software always generates an internal line feed when it does a carriage return. Some software does not. You are safe answering YES to this question. The most that can happen is that you get double-spaced lines which are easy to read but scroll off the screen fast. If you answer NO, you may find all your data printing over itself on one line.

You will be asked to enter your first name, last name, and the location you are calling from. If you have not been on the system before, your name will be added to a log file. You may be asked for specific information about your terminal equipment. The system may ask if you can use lower-case characters, how many characters you can display in a line of data, and other information. This will be kept on file with your name, and the next time you sign on, the system will know you and treat you accordingly.

Certain systems may interrogate your terminal software to determine if you are using a program with special features the system recognizes. An example of this is the Forum 80 message system and ST80-III terminal software. Each copy of ST80-III can automatically tell the Forum 80 what its unique software serial number is for purposes of sure recognition during sign-on.

The next thing you see on your screen will certainly be the

welcome message and system announcements. The announcements may be any information of interest to the operator and users. They will also probably include information on the control characters or special commands the system accepts. Some systems have expanded files of historical and system configuration data. You may be interested in browsing through later, but for now let us move on to the place where the business is done, the command line.

Commands and Control Codes

Most of the electronic message systems (CBBS, ABBS, and others) use similar "top-level" commands. These are the first command choices you will make. Since there is no formal standard, we will examine a typical command line and see what the more common commands do. A typical command line might be:

FUNCTION:B,C,D,E,G,H,K,O,Q,R,S,X,?

The command line wants you to enter a letter to tell the system what to do. If you are confused or you do not remember the commands you want, you can always enter either H (Help) or ?. Either one will provide you with help in understanding the commands for the level of the system you are on. The H command will usually provide more detailed instructions. Here is what some of the other commands mean:

B Entering this letter will display the bulletins which you might have seen at sign-on. A longer historical list of bulletins might also be displayed.

C This is the "Case" switch. It switches operation of the system between upper case-only and combined upper/lower-case. This is a toggle operation. Each time a C is entered the system is toggled from one mode of operation to the other.

D The D command toggles between full- and half-"Duplex" (echo/no echo). Echo is the normal or "default" operating mode (full-duplex).

E This tells the system you want to "Enter" a message. You will be asked to specify to whom the message is going and what the

subject is. You should do this in upper-case and in a standard form (for example, TRS-80, not "trs80") so others who use a system scan for messages of interest will not miss yours.

G This says "Goodbye." This initiates the sign-off to end the session. (Special Note: A "T" for "Terminate" is used by Forum 80 systems to end the session. Sending a Forum 80 a "G" puts you into the graphics mode of operation which can greatly confuse non-TRS-80 terminals. A sign-off command should be sent to all systems. You should never simply hang up because the system may wait for input from you for several minutes before recycling. During that time it would not be available for other users.)

K The K command is sent to tell the system you want to "Kill" a message. Often messages are "locked" with a password so you may not be able to kill messages you did not originate or for which you do not know the password.

O This stands for "Other" systems. This normally provides a detailed listing of other message systems around the country. The accuracy of the list varies with the kinds of system you are on and the concern of the SYSOP who must make all of the entries and corrections. Generally, a Forum 80 system will have an accurate listing of all other Forum 80s. The only other generalization that can be made is that a large percentage of the telephone numbers listed on any system are no longer active—despite the most dedicated efforts of the SYSOP.

Q The system will present a "Quick" summary of messages including only information such as who the messages are from and to whom they are addressed.

R This command allows you to "Retrieve" a message from the file for reading. You will be asked for the message number, so you probably will have to execute the Q or S instructions first. Various suboptions under the R command may allow you to retrieve multiple messages sequentially or messages you flagged for retrieval on a previous scan.

S This commands a more detailed "Summary" including the subject, date entered, and possibly the length of the message. Often, scanning can be done selectively. You might wish to scan for all messages with your name in the TO line or all messages with Apple

in the SUBJECT line. Directions for performing a selective scan are usually contained in the Help file. Some systems have so many messages on file that you might also be asked what block of messages (by message number) you wish to scan.

X An X command indicates an "eXpert" mode of operation. The system skips long prompts and "hand-holding". It assumes you are an experienced system user.

If the system you check into is anything but a Forum 80, one of the notices that may appear in the sign-on and introduction was probably: CONTROL CHARACTERS ACCEPTED BY THIS SYSTEM.

This is accompanied by a quick list of control characters and their functions. Control characters are important to the smooth operation of many systems because they provide an executive level of control. The ability to transmit control characters is advantageous for this reason. Forum 80s do not recognize control characters because the early TRS-80 terminal software did not produce them. The most commonly used control (CTL) characters are:

CTL C (Cancel) This command may either cancel the current line being sent or the entire message being currently printed.

CTL K (Kill) This kills the current function (Scanning, Reading, and so on). Operation may return to the command line or to the next logical function if you are down in a subsystem.

CTL N (Nulls) Send nulls after a carriage return.

CTL S (Stop) This control is also called XOFF, it stops the distant system from sending you any more characters. This is handy if you want to study something on the screen, make notes, or answer the doorbell. On some systems another CTL S is needed to restart transmission. Other systems will resume when any new character is received from you (hit any key). The standard response is CTL Q.

CTL Q (XON) This control code is used by many systems (including Source and Compuserve) to signal a restart of transmission after a CTL S.

Several other control codes may be used for purposes such as flagging (CTL F) messages for later retrieval and cancelling all functions (CTL X) causing a return to the command line.

The commands and control codes listed above are the ones in most general use. Many systems have additional commands available. Also, certain commands may have different meanings at different program-function levels. The S command, for instance, means "scan" at the top (initial sign-on) level of the operating program. But during the formulation and entry of a message (a lower-command level) the S may cause the message to be "saved" to disk. Similarly, an E command may become "edit" instead of "enter." This reuse of the same letter command is usually not confusing because detailed prompt are provided to the user. The "?" or "Help" request will usually bring a clear explanation of a command's meaning at that functional level.

Each type of message system has its own particular short cuts and operating conveniences. Some systems allow you to "stack" your commands or responses all on one line to speed entry. This presumes the user knows what questions the program will ask. The questions the program asks can be answered before they are displayed if the answers are separated by a semicolon. As an example, experience may have shown that a CBBS asks the following questions in the following order: IS THIS YOUR FIRST TIME ON THE SYSTEM?, FIRST NAME?, LAST NAME?. All three questions can be answered on one line. A typical stacked entry line might be:

(computer sends)	(you reply)
FIRST TIME?	N;FRED;RALPHSON

Stacking commands can be used in other situations. You may have learned that the system you use responds to a retrieve command with the questions: MULTIPLE RETRIEVAL?, and then MESSAGE NUMBER?.

Your stacked input line might be: R;N;22 which tells the computer you want to retrieve the single message numbered 22. The single "N" above in both cases is shorthand for "no." Similarly, a "Y" represents "Yes."

Some systems cater to their regular users. Frequent users can often arrange with the SYSOP to provide for special privileges

such as private message listings, auto log-on numbers, or access to the master files.

Many systems have a program upload and download feature. This turns the message system into an electronic program exchange. Various utility and game programs are available free of charge to system users. Users are encouraged to contribute personal or noncopyrighted public domain programs for the use of others. A smart terminal program is required to save the received program for later reuse. Many message systems allow the user to run resident programs. This is more of a classic computer utility function, but it is a valuable service for persons using a standard terminal or a microcomputer-based terminal with a RAM space too small to run the desired program.

Special interest message systems are also popular. These special interest systems range in type from those dedicated to technically-oriented users of the CP/M operating system to those designed for persons interested in genealogy. Other systems dedicated to avionics, various kinds of medical education, photography, electronic video systems, amateur radio, engineering, and computer game-playing have all been established.

Using the special features of microcomputer-based message systems is not difficult. A great deal of effort has been expended in making systems easy to use. A careful reading of the HELP commands and a little practice will lead to a very enjoyable data communication experience.

SETTING UP AN ELECTRONIC MESSAGE SYSTEM

Establishing an electronic message system can simply involve the combination of some readily available hardware and software. In order to determine what hardware and software are best for a particular application, two questions must be answered: 1) What is the purpose of the system? and 2) What hardware is available?

Determining the actual purpose of the system involves a study of what services are needed, how many users and messages may be handled, and any special features such as privacy or different data alphabets desired. If cost is a major factor, the the use of available hardware may be a big consideration. Large message

systems can use a great deal of disk space and high-capacity disk systems can expensive. Fortunately, some less expensive combinations of hardware can be used.

Remote Operation

The simplest sort of message system can be referred to as *remote operation*. In this kind of system, the microcomputer is made available for remote entry and operation through the RS-232-C port. The serial port is, in effect, placed in parallel across the local keyboard by the software. Data coming in from the serial port is displayed and acted upon just as if it were entered locally. Remote operation can be extended any distance by the use of modem and the telephone lines.

During remote operation, programs in RAM or available on disk can be run as long as they do not contain machine-language subroutines incompatible with the remote operation program or do not require local physical actions such as the changing of disks. In the most elementary form, messages could be left on a system like this in the form of BASIC programs full of REM or PRINT statements. Programs could be LISTed or RUN to read the text.

Many modems, including the Microconnection and LYNX, have an auto-answer function which will at least answer the telephone under software control. TRS-80 Model I, II, and III microcomputer systems are all easily provided with a remote operation feature.

A simple program is available to allow remote operation of the TRS-80 Model I. The COMMUNICATOR from Instant Software is a cassette-based program that provides complete remote operation of a TRS-80 through the RS-232-C port. The program resides in high memory and does all the interface necessary to serve as a computer utility including echoing received characters. It does not include the routines needed to answer and hang up the telephone line through the modem. A device such as the Hayes Microcomputer Products SmartModem has answering and hang-up routines built into its operating system.

Two other programs are available for the TRS-80 Model I which allow message-system operation with the most simple Level II 16K keyboard-only system. Both require the use of the Microconnection modem device with the Autoconnection option.

SUPERHOST provides full remote operation with telephone answering and hang-up under program control. MINI-MSG is a more complete message-system program which actually functions as a bulletin board. Users are greeted on sign-on and can read all messages or enter messages of their own. It runs completely in RAM and stores all messages in RAM. No disk or cassette storage is needed. RAM-only storage provides a practical limit on the size or length of the messages available, but a 16K system can serve the needs of a surprisingly large community of users if messages are kept brief. RAM-only storage also means all messages are lost if the power fails. The program has limited, but effective, HELP commands. This program would be ideal for any small business, school, or club that would like to operate its own electronic message system at minimum cost. SUPERHOST sells for about $30 and MINI-MSG for $50 from the Microperipheral Corporation. Both programs address the Microconnection ports and will not work with other modem devices.

Lance Micklus has written a very practical message-system package which includes a message bulletin board, runs in a 48K Model I or Model III TRS-80, and can address the standard and nonstandard TRS-80 ports. The package consists of two separate programs. The first is a machine-language program allowing remote operation called ST80-X10. A separate BASIC language bulletin-board program called Communications Center (ST80-CC) works with ST80-X10 to do the actual message-system functions. The BASIC language message program has options for various levels of entry by users. Users may be required to provide a system password for full entry into the message storage or program area. "Guest" users may be allowed limited entry without a password. They can, for instance, be limited to leaving messages only to the SYSOP. This kind of system has good commercial applications for taking sales orders or selling information such as stock tips. It could also have good uses in training and education. The ST80-X10 and Message Systems programs will run on the TRS-80 Model I and Model III systems. ST80-X10 costs about $50 and ST80-CC costs about $100. They are available through the Small Systems Business Group. A simple message program, ST80-PBB, is available on tape or disk.

Trying to describe sophisticated electronic message systems in detail is somewhat like trying to describe the patterns of flames in a fireplace. You know they are warm and complex and beautiful, but they keep changing. The people writing message-system software do so mainly for the love of it. They try to keep commonality and to prove good documentation, but they are very similar to artists who must add just one more brush stroke to their work. They often wind up repainting large sections of canvas. However, those conditions not withstanding, in this section we will look at several sophisticated message-systems, describe the hardware needed to run them, and tell where they are available.

Several of the best-known message-system packages are not for sale as money-making products. They are available for a cost which does not reflect the work that went into them. But they are also available without the "handholding" support expected from a software distributor selling a major program. The authors might also want some control over their use. Let us look first at the CBBS software.

CBBS

The CBBS program written by Ward Christensen runs under the CP/M operating system. It comes on two eight-inch floppy disks which contain many thousands of lines of machine-language code. The CP/M operating system must, of course, be purchased separately and be fully integrated with the hardware.

The hardware requirements to run the system are an S-100-based microcomputer system with a Z-80 or 8080 CPU, at least 24K of RAM, a disk system, and an auto-answer modem. An integrated modem such as the Hayes or PMMI is preferred. A minimum of 70K of disk space is required, and much more is really advisable. Although the program is written to run under CP/M and that operating system is certainly not restricted to S-100 bus machines, the program is designed to run on an S-100 system.

This program may require some degree of experience or

patience to bring it up with different hardware configurations. The price is kept low ($50) because the system developers do not have the time to help every person who might try to bring a system on line. Common hardware configurations using the PMMI or Hayes modem devices can be brought up quickly, but bringing any special requirements or configurations on-line may require a good knowledge of CP/M and machine-language code.

Forum 80

The Forum 80 system is referred to as a "network" by its creator, Bill Abney. The various Forums are not connected electrically (although message transfer between systems is a real possibility), but they are connected in commonality of software and exchange of information on a regular basis.

The Forum 80 software is being sold through the Small Business Systems Group. It will operate on the TRS-80 Model I or III. A system requires three disk drives on a Model I and two disk drives on a Model III. The microcomputers also have to be equipped with 48K of RAM, an RS-232-C interface, and an auto-answer modem.

The Forum 80 software package is very professionally done. It has a fully capable message system and other special features such as space for special electronic newsletters. The Forum 80 software package sells for about $350.

ABBS

The Apple Bulletin Board System software is certainly the most widely used electronic message-system program. The software is easy to bring on-line and doesn't require extraordinarily large disk capacity. The hardware package for an ABBS typically consists of an Apple II Plus or Apple II with AppleSoft in ROM, a Hayes Micromodem II, and at least one disk drive. An additional disk drive is needed if any optional program features are used. The software can also support large-capacity hard-disk storage systems.

The ABBS software is supplied on two separate disks which contain various system configuration routines. The user is, in effect, taken by the hand and led through a series of steps which customize the software to the available hardware and desired operating parameters.

There are many optional features available for the ABBS system software. Program exchange is one of the most popular features. Users of the message system can download programs to their own terminal and save them for later use. They can also upload programs into the message system so that others can borrow them.

Another valuable option includes special user "conferences." These conferences are actually message subsystems for special-interest groups. They can be given any desired names and are available as an option from the command line.

Craig Vaughan markets the ABBS software and other related communication programs through Software Sorcery, Inc. The initial software package sells for about $70. Options can run the total cost to over $150, but this price is low considering the quality and ease of use of the software. Vaughan also has programs with greater security options for corporate users and communication programs for users of the Pascal programming language.

Peoples' Message System

The Peoples' Message System is a bulletin-board system using software written by Bill Blue. The software is designed to run on the Apple II Plus computer equipped with the Hayes Micromodem II. It can be operated on as little as two Apple minidrives, and it is upwardly expandable to eight-inch drives or a hard-disk system. The primary system can accommodate up to 116 messages, but the system will automatically purge old messages as new ones are entered. Additional messages can be stored if more disk space is available.

A separate file for news or system bulletin is available, as is a features section for providing on-line articles, advertising, club news, or any other features of interest.

Bill has even included an "obscene filter" in this system. This filter consists of a file on one disk containing all of the words or phrases the operator does not wish to be displayed. Each message saved to disk is checked against this file, and if any matches are found, the message will not be saved.

The Peoples' Message System has many other features such as the ability to assign "accounts" to frequent users and to allow users to upload and download programs. These features are in

extensive service. This program comes individually configured for each installation and costs between \$200 and \$300, depending on options.

North Star

The Microstuf Company has a bulletin board system which is written in North Star Basic. The program package, called "Bulletin Board," is actually a series of programs including the message program itself, a log-on program, menu and help programs, and a utility program to pack the disk files. The package looks like one integrated program to the user and is easy to install on a Horizon computer. Selected users may gain access to BASIC language programs in the computer and run other programs, but only under control of the Bulletin Board package.

Versions of "Bulletin Board" are available for the PMMI and Hayes integral modems. A version operating in Microsoft BASIC under CP/M is also available. The "Bulletin Board" (with its accessory remote access program) sells for about \$100.

PET & CBM

Fred Hambrecht deserves credit for putting the first PET Bulletin Board System on the air. The PBBS software can run in a 16K PET or CBM with one disk. Either the TNW 2000 or the TNW 488/103 serial-port devices are required. The message-system software is available for about \$25 with extra cost options available.

Microcomputer-based electronic message systems of all kinds are a new phenomenon in the world of communication. They grow, change, and have impact on the world around them with amazing speed. Their uses are only as limited as the imagination of the people who program and operate them.

One of the most humanitarian of all message-system uses is as a communication medium for handicapped individuals. The next chapter deals with message systems for the deaf.

9

Communications for the Deaf

Perhaps no group of people can benefit more from data communication systems than the deaf and the hearing-impaired. Deaf people need a way to communicate that is not based on sound. Unfortunately, the technologies first developed in the communication revolution—telephone and radio—depended completely on sound. Throughout the twentieth century, deaf people have been culturally isolated because of the importance of sound to society.

If you are deaf, how do you call your boss to say you will be late for work? If you are deaf, how do you call the police or receiving warning of severe weather conditions or utility outages? Quite frankly, you often do not.

The deaf have been using electrical data communication equipment since the 1960s. Ironically, this early establishment of a transmission method has actually worked against their participa-

tion in data communication developments based on the microprocessor.

TTY COMMUNICATIONS

In 1968, an agreement was negotiated between the Alexander Graham Bell Association of the Deaf (in Washington, D.C.) and American Telephone and Telegraph Company (AT&T). It was agreed that AT&T would provide deaf persons with surplus Model 15 teleprinters. (Teleprinters are also known generically as TTY or "teletype" machines–Teletype is a trademark of the Teletype Corporation.) AT&T and the communication carriers served as a source of these machines through the late 1970s.

Most of the machines were used by deaf persons in a point-to-point mode. The surplus machines were installed in homes, schools, businesses, and even some police stations. This was personal communication. Deaf persons could actually call their friends and "chat" by long distance or call schools or civil authorities for help. Estimates placed the number of deaf people using data communication at over 77,000 by the end of the 1970s. (There are about 2.4 million deaf people in the United States.)

The technology used in these communication schemes predates World War II. The teleprinter I/O port is a direct-current interconnect to electromagnets which push on typebars attached to the typehead. The coding of the DC line is done in a 5-bit code referred to as Baudot or Murray code. Timing for the reception and transmission of the bits is done by DC motors. The speed of transmission is typically 60 WPM or 45.45 baud.

The modem used to send the data over a telephone line is called a Weitbrecht modem. Only two tones are used over the transmission path–1400Hz mark and 1800Hz space. In the Weitbrecht modem, tones are only transmitted for the duration of the character so that the line is available for transmission in either duration between characters. In practice, the mark tone is held for as long as a second before it is dropped to provide continuity while continuous strings of characters are transmitted.

Deaf communicators have come under various strong pressures in the 1980s. Time and technology seem to be working

against them. Their mechanically complex teleprinters are wearing out, and parts are difficult to find. Technology has turned in another direction, and ASCII coding at speeds of 300, 600, and 1200 baud has become standard. Services such as information utilities, which would meet many of the needs of deaf people, do not usually interface with their machines. Many deaf persons simply cannot afford to buy a microcomputer so they can rejoin the data communication revolution. Those who can afford new equipment hesitate to leave their friends with 60-WPM Baudot-coded devices behind.

One obvious solution would be to provide microcomputers serving as message systems and as terminals with a Baudot code capability, along with their ASCII capability. We have seen several programs using translation tables which can essentially turn an ASCII-coded character into any other ASCII- or non-ASCII-coded character instantly. This translation is easily done in software. But the major problem is not coding; it is speed. The designers of the integrated circuit devices in use today did not provide for speeds below 110 baud. Some special hardware and software magic must be worked to meet the lower-speed communications standard for the deaf.

AMRAD

One organization has been working hard to produce this magic. The Amateur Radio Research and Development Corporation (AMRAD) is a nonprofit group of volunteers who have been working to find ways to bring deaf data communicators into the new wave of technology and to interface microcomputers with communication devices the deaf have.

In 1978 AMRAD established the fourth electronic message system to be operational on the telephone lines in the United States. It was the first bulletin board system to also be accessible over amateur radio through the use of 5-bit Baudot-coded signals. A similar bulletin board system, known as the Virginia TTY, was set up separately to serve the deaf community in the Washington, D.C., area. Later, a more sophisticated system called HEX was brought on-line; it provided for ASCII-Baudot interchange and disk storage of messages which the smaller system did not have.

HEX is available to deaf callers using TTY standards at (301) 593-7033.

AMRAD also has an active project underway to interface microcomputers acting as terminals with communication devices for the deaf. The Apple II computer is easy to interface, and several demonstrations of this capability have been made. A simple interface device consisting of two ICs, some diodes, one transistor, and a handful of resistors is connected to the Apple II game I/O port.

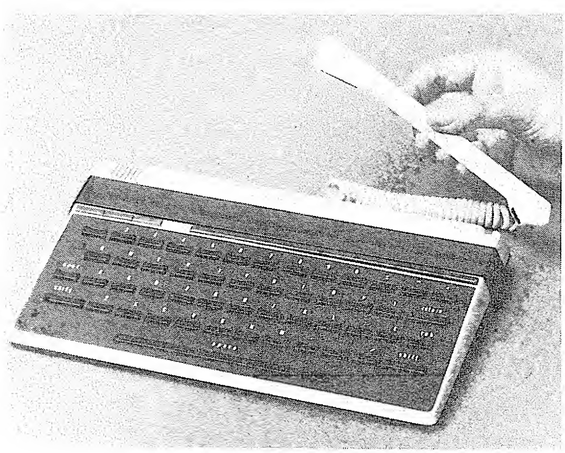
The TRS-80 is more difficult to interconnect. Parts alone cost $70 or more to build a proper interface. Work has also been done on PET/CBM, AIM 65, and other microcomputer systems.

Anyone interested in interfacing a microcomputer to communication systems for the deaf can contact AMRAD for more information at 1524 Springvale Avenue, McLean, Virginia 22101. Please remember, AMRAD is a volunteer, nonprofit organization. The organization has no products to sell, and they can provide information only on an individual basis.

INFONE

Novation has produced a portable combination terminal/telephone aimed at the deaf market called Infone (Figure 9-1). It has a store-and-forward message capability and is compatible with both ASCII

FIGURE 9-1. The INFONE Device from Novation.



and Baudot code. It also has the capability to "speak" in synthetic speech for blind users. Novation has designed Infone around a special "modem on a chip" which aids in keeping both size and cost down. The terminal can communicate through a direct connection to a standard modular telephone jack or via an optional acoustic modem.

The small desktop unit has a 2000-character memory expansible to 7000 characters for message storage and a 250 word vocabulary for synthesized speech. It uses a linear, predictive coding chip from Texas Instruments.

Technology will continue to help deaf communicators in many ways in the coming years. It is already possible to put a microcomputer such as the TRS-80 Color Computer on-line as an ASCII terminal with prepackaged hardware and software for around $500. As the value and popularity of data communication services grow, the price of mass-produced terminals will certainly drop even more. Deaf data communicators should be allowed to enjoy the many services that come with microcomputer-based data communication systems.

10

Large-Scale Communications Networks

This chapter will attempt to define and explain three different kinds of large-scale communication networks. These networks have many similarities in that they all consist of computers, transmission media, and private customers. Each kind of system emphasizes a different portion of the operation.

The three kinds of networks are information utilities, value-added carriers, and alternative telephone carriers. Information utilities place emphasis on their computing power and the size of their data base. Users call in over telephone systems and (as we shall see) over value-added carriers, in order to use the computer. Information utilities include services such as the Source and CompuServe.

Value-added carriers often provide the links between the customers and many information utilities or computer utilities.

They consist of local and network communications computers which move the users' data around efficiently and at high speed. The data from many customers can be packed into dedicated transmission systems at low cost. Additionally, the network computers detect and correct errors on the transmission link, buffer, and convert data arriving in various formats and speeds. The function of the network computers is invisible to the user. The emphasis is on carrying the data, not proceeding it.

Alternative telephone carriers provide competition for the large long-distance telephone carriers which have had a near monopoly on telephone communications in the United States for many years. They use computers to route calls. A pushbutton telephone actually serves as a terminal device, which tells the system computer who you are and where you want to talk. The emphasis is again on carrying the communications, but usually in terms of voice circuits.

Let us look at specific examples of each kind of system and see how they are related.

INFORMATION UTILITIES

Two major information utilities are available for the use of the general public in the United States. Many other systems are operational and highly successful in industry, education, and government, but the Source and CompuServe bring mainframe computing power to private users. Both systems have developed into what might be considered electronic newspapers, but with different formats.

The two systems have been competing in a very hectic race, and the public has been the winner. Each system brings new features on-line almost daily, so it would be foolish to try to specifically list their services. It is possible, however, to generalize about the kinds of things you will find on both systems and the operating differences of each.

CompuServe and the Source both offer access to individuals over standard local telephone lines. In most cases, a local call is made to the regional terminal input point (TIP) of a value-added carrier. The value-added carrier mixes the individual's data in with

that of possibly hundreds of other users and moves it over high-speed circuits to the large computer cluster of the information utility.

The user may have two sign-on sequences: one for the value-added carrier and one for the information utility itself. This is a simple process involving the typing of a few characters and letters. Calling an information utility is as easy as calling a local ABBS in most metropolitan areas of the United States. CompuServe provides its own direct lines to many major cities. In smaller cities and rural areas, a long-distance call may be necessary to reach the nearest entry point. This factor should always be considered when joining any information utility system, because the coverage patterns are not identical, and long-distance bills can add up quickly.

After sign-on, the user is greeted with announcements about the system. They are very similar to the headlines on the front page of a newspaper. They may advertise new features, system news, or important national news. Both systems are similar in the kind of menu that greets the user at sign-on. Users can select categories and subcategories of information from menus and enter the number of the feature they would like to see. This menu-driven format is easy for new users to use. They are led through a series of increasingly more specific programs until they get the desired information. CompuServe provides the users with the option of skipping menus through the use of direct GO commands. These commands move the system to the desired electronic page of data instead of working through various levels of menu selections. This option speeds the service for more experienced users.

The Source has a menu-driven format for new or infrequent users, but the system also uses direct command statements. If the user leaves the menu and enters the command level, a prompt (>) will appear and the system will wait for the user to tell it what to do. Menu-driven systems are friendly for new users, but command-driven systems allow experienced users to get powerful work done quickly.

Both information utilities do a good job of presenting national news stories. Again, the more flexible entry format of the Source allows experienced users to search for particular news stories by keywords, but the menu-driven format of CompuServe allows easy browsing through lists of important stories which might otherwise be missed.

Other features on the systems include business news, stock reports, classified ads, seasonal specials, shopping (via the catalogs of several different manufacturers), computer games, and electronic mail. The electronic-mail system of the Source is very flexible in that it allows forwarding of mail, replying, and other features. CompuServe's EMAIL is less flexible. Source mail will accept an almost constant dump of characters from a prepared file. EMAIL requires a terminal transmitting prestored messages to respond to line-by-line prompts.

Both services provide the user with the ability to use the system as a more traditional time-shared computer. Various programming languages are available, and large disk and RAM storage can be used at low daily rates. Programming aids and utilities such as text editors are available for persons using the higher-order languages. Specific programs can be written on line and saved, or any of the many programs in the system library can be selected and run. Use of the library programs and storage are extra cost options.

Each system is actually a cluster of computers. The Source operates on a group of prime computers located in Maryland. The CompuServe service runs on a network of over twenty Digital Equipment Corporation mainframes in Ohio.

Most users will take advantage of the nonprime service hours of these systems. Nonprime service hours for CompuServe are 6:00 P.M. to 5:00 A.M. your local time and all day on Saturday, Sunday, and holidays. The Source has similar nonprime hours. Rates for nonprime-time service run between five and seven dollars an hour. Prime-time service (during the "work day" in your local area) can go higher than $20 an hour.

Terminal software for the most popular microcomputers is available from both the Source and CompuServe. CompuServe has cooperated closely with Radio Shack in integrating the Radio Shack Color Computer and other TRS-80 products into their system. The Videotex series of programs for the TRS-80 and Apple II microcomputers allows transmission of control codes and provides recognition of special cursor-positioning and screen-formatting commands. The program for the Color Computer allows transmission of messages prepared off line.

The future of these information utility systems seems unlimited. They will continue to bring larger data bases and more unique features on line. They have many options opening in the

future for different ways of reaching their users. These options could include integration into cable television systems and possibly even direct satellite broadcast. Some microcomputer system users may seriously consider the balance between buying and maintaining a large microcomputer system and paying five to seven dollars per hour of evening use (plus optional services) for the almost unlimited computing power of a computer/information utility.

Before we leave these systems completely, there are two other systems you should know the names of, if only because they come up so often in reference. They are PLATO and the Arpanet.

PLATO is a time-shared system serving hundreds of colleges and universities throughout the country. Its main trust is toward education and the use of computers in instruction. Hundreds of thousands of college students and faculty members have learned about some of the practical applications of data communication through PLATO. PLATO has served as a developmental base for many of the ideas we now incorporate in our data communication systems.

Similarly, the Arpanet developed data transmission techniques which are now the heart of modern data transmission systems. "ARPA" is part of the acronym DARPA which stands for the Defense Advanced Research Projects Agency. DARPA is a small military agency which works at the leading edge of technology. The people in DARPA saw the potential for data communication and established a developmental interactive computer network many years ago. The network is now administered by the Defense Communications Agency and is used by military agencies and contractors.

This network pioneered the packet-switching technique, which breaks strings of data into packets of information that are easily moved around in the transmission system. These packets can be routed in several different ways. Routing is done to insure the best utilization of all data trunks and balance between them. When people write or talk about the Arpanet technology, they usually mean packet switching. Packet-switching techniques are now being used by the next kind of communication network we will look at, the value-added carriers.

VALUE-ADDED CARRIERS

Let us begin with a quick primer on digital communication network-switching techniques. You need to read through it if you are going to understand the packet-switching techniques used by value-added carriers.

Picture a large data communication network as a telephone system. The major differences stem from the fact that data networks carry digital signals instead of analog voice messages. Each network has many subscribers linked together through a system of switching centers. In the telephone system, the switching centers switch calls based on the numbers dialed. The digital switches used in data networks may use different methods of switching data.

There are three kinds of digital switches: message switches, packet switches, and time-division circuit switches. Message switches came first, and they are exemplified by the CBBS, ABBS, and Forum store-and-forward electronic message systems. A complete message comes in, is stored, then forwarded at the appropriate time to the appropriate recipient. The storage part of this process can take a long time, and store-and-forward message switching can be a slow way to send a lot of messages.

The next technology to be developed was packet switching. A "packet" of data is actually a string of characters. Packets can vary in size according to the design of the specific system. Some systems may use packets 2000 characters long. Two-thousand characters may sound like a large message instead of a compact packet, but it is small compared to the huge streams of data transmitted by large computerized systems.

In packet switching, a block of data has addressing and possibly routing information added to it, so that it can travel independently through an interconnected network of switches to its destination. Packets travelling independently through the system may arrive out of order at the destination, but they can be rearranged by the network's processors into the correct sequence. Some delays may take place while packets are being formed and addressed and while they wait for each other to get through the system.

A third alternative in digital switching is called time-division circuit switching. These systems use very small blocks of data (even just one character) as their transmission element. They avoid adding addressing and routing information to this small data block by the use of signalling channels, which are separate from data channels. As the character or data block moves to a switch, instructions go out on a separate line as to what to do with it. The coded instructions are very simple, and the data channel is kept clear of all overhead. Time-division circuit switching is a newer technique which has not been developed as fully as packet switching.

Most of the value-added carriers today use some variation of packet switching. Two packet-switched carriers are commonly used to enter information utilities: TYMNET and GTE Telenet. These carriers provide efficient transfer of data between large computers and between computers and customers' terminals. The users make a local telephone call to enter the data network, although large users will probably be connected to the network through dedicated lines.

Each service has entry facilities in about 300 cities in the United States. Service is concentrated in the urban areas of the East and West coasts. Many rural areas may be without value-added carrier services until satellite transmission facilities become available. The services have extensions into Canada, Japan, and Europe through international agreements.

The "value-added" service provided by value-added carrier consists of two primary areas: error detection and correction and matching of different kinds of data communication formats together into a common network.

Error detection and correction removes and corrects "bad bits" of data that were damaged or dropped during the transmission. The process relies heavily on statistical routines to check literally millions of bits on a network every second.

The transparent interface feature of these networks allows many different kinds of terminals using many different electrical standards and coding schemes (there are probably over a dozen in common use) to talk together without modification. The network's translation features are invisible or transparent to the users.

The information utilities make use of TYMNET and Telenet to carry the data between their mainframe computers and the individual users. Many casual communicators use these highly sophisticated data transmission networks every time they dial an information utility, yet they are not aware of the functions the utility provides.

ALTERNATIVE TELEPHONE CARRIERS

Once there was only a couple of big telephone companies and a smattering of small ones in the United States. They had the country carved into different geographical areas, and all services they provided were billed according to tariffs set down by federal and state regulatory agencies. Certain long-haul communication carriers were chartered to interconnect the regional companies into a national and worldwide telephone system. All these companies were in a very well-regulated and protected world. This monopoly led to the establishment of the first and best (both in terms of quality and availability) telephone system in the world.

The monopolistic portion of the communication structure was eventually challenged by companies who wanted to connect their own equipment to telephone circuits. Private companies finally won the right, through long and tough court fights, to compete in various parts of the telephone market. Now, many types of FCC-registered equipment are available which can be directly connected to the telephone lines, including modems.

Similarly, alternatives to the standard telephone carriers became available; but at first these services were only economical for large corporations, who used great quantities of telephone service and could see the benefit of saving a few cents on every call. This alternative form of telephone service became available to private individuals in many parts of the country in the early 1980s.

An alternative-carrier network is really a system of voice circuits controlled by computer billing and switching systems. The voice circuits ride over microwave or satellite systems. The voice circuits ride over microwave or satellite systems that may

be owned by the alternative carrier of leased from yet another communication company. The circuits travel in large groups, perhaps with several conversations combined on each circuit through multiplexing techniques. These groups of circuits go from city to city. In each city, they connect to the existing local or metropolitan telephone system. This technique allows the alternative carriers to use high-technology, low-maintenance links between cities, while the local telephone companies provide the less glamorous but very vital local service to homes and businesses.

Users of alternative telephone services make calls in a two-step process. In the first step, they dial a local number to reach the alternative carrier's entry equipment. They then receive another, easily recognized, dial tone. In response to this signal, they enter a series of numbers using the push-button telephone dial. These numbers represent the customer account code. The area code and number of the party being called are also entered at this time. The alternative carrier system then routes the call over its own circuits to the city being called and redials the calls into the local telephone system. All this is done at a price that may be considerably less than normal long-distance telephone service. Obviously, such a service would have great appeal to persons who want to dial into electronic message systems across the country.

There are two major alternative carriers offering service to private individuals. MCI Telecommunications and SP Communication offer two competing services. MCI uses the trademark named Execunet, but they also refer to themselves as "The nation's long distance phone company." SP Communications calls their service Sprint.

Both services have similar (but not necessarily identical) billing schemes. Individual users pay a basic monthly service or subscription charge of around $10 to establish an account. They are then charged for each call according to the time of day, distance, and number of 6-second time blocks used. Calls made during the daytime hours can cost as much as 30% less than those made on the more traditional carriers. The differences are less at the late night or weekend rates.

A prospective user of these alternative telephone systems has several factors to weigh. The primary factor is the geographical area of coverage. Many cities and rural areas are not covered by this service. Remember, both the originating and terminating

areas must be covered by the carrier for the call to go through. The monthly service fee must be considered as part of each call made. During some times of the day and weekends, the savings on a three-minute call across country may be as little as twenty-five cents. The savings may not offset the service charge unless many calls are made. Additionally, the standard telephone carriers may be able to provide Wide Area Telephone Service (WATS) rates if calls are often made to a specific area. All of these factors must be considered, but generally, if frequent calls are made on weekdays to metropolitan areas, these alternative telephone services may save you a considerable amount of money.

One final note: The quality of the circuits provided by these carriers may vary. They try to meet the highest standards of the industry, but they lack the alternate routes and widespread maintenance base needed to allow constant line checking and maintenance. Most data communicators have found the service perfectly adequate for 300-baud service, but other have reported more frequent errors when using these services for data communication. Most problems can be worked out with the carrier.

THE NATIONAL NETWORKS

All these network systems represent a great potential for the transfer of information and the interchange of ideas. Their growth will continue to have a great influence on the way we learn, play, work, and shop. The future of the industrialized countries of the world will be greatly influenced by communication networks. In the next chapter, we will look at the future of data communication systems and networks.

11

The Future

Predicting the future accurately can be a very difficult task. In the early 1950s, predictions were made that personal helicopters were right around the corner and that commuters of the 1970s and 1980s would take off from their backyards and land on their office roots. Scientists in the 1960s predicted colonies would be in space by the 1980s. Unfortunately, these things have not happened. But the history of technological forecasting has many more examples of unforeseen developments than of wrong guesses. The things we can see usually happen in one form or another, and they happen faster than we normally think possible. The things we cannot see far outnumber those we can. In this chapter, we will outline some developments that are reasonably easy to predict—which probably means they are right around the corner.

What developments will we see in terminals over the next few years? The first development (demonstrated in the late 1970s) will be voice recognition and voice synthesis. Terminals will be able to recognize and act on voice commands—much as the special-function keys today perform extended functions from the pressure on one key. More complex terminals, able to take dictation and provide highly reliable translation from spoken to written form, will be available by the end of the decade. Terminals and peripheral devices will also "talk" to their users. Perhaps they will only say a few standard words at first (perhaps "READY" will be the most common word a terminal speaks), but soon terminals will be able to speak every word as it appears on the screen.

But even spoken interfacing has its limitations. The human/machine barrier will eventually become totally transparent to the user. It is certain that an effective physiological or biocybernetic link between people and terminals will be developed. This link may not be as sophisticated as a mind implant, a favorite topic of science-fiction writers, but it may be just as effective. Various kinds of sensors attached to the body can exchange stimulus messages with a computer. These stimuli can be translated by the operating system or the mind into meaningful information or physical shapes. Two variations of this process have already been successfully demonstrated.

Pilots of high-performance fighter aircraft may have their vision blurred during high-speed maneuvers, yet they still need to receive vital information. Warning horns and recorded women's voices have been used, but these sounds interfere with important radio messages. One system allows computers in the aircraft to communicate with the pilot through the use of a pulsed pressure cuff—similar to a blood-pressure cuff—on the pilot's thigh. When the aircraft is near the edge of its performance envelope, the cuff begins to pulse with compressed air controlled by the aircraft's computerized flight system. The worse the condition of flight, the faster the rate of pulsation. Pilots quickly learn to judge aircraft performance by the pressure on their leg. This is an easy and sure physiological communication link between a human pilot and a computer.

Another more complex example involves television images which are "seen" by blind people through pressure on their backs. The images from a television camera are changed to digital pulses; the pulses are fed through a computer which activates a device containing hundreds of tiny pins set in a rectangular pattern. This pin system can reproduce the television image by placing pressure on the pins to form an outline of the image. Blind persons can be trained to "see" the outline when the pin device is placed on an open stretch of skin like the back. Continued training can lead to very accurate perception of objects in a kind of reprogrammed vision. This is another stimulating example of the kind of bio-cybernetic interface that can be developed with a little imagination. Other, more capable, links will certainly follow. The flexing of fingers during the operation of a keyboard is a learned physiological act, but there may be more effective physiological ways of receiving and inputting information to a computer.

The real advantage of bioelectronic or physiological interfaces is size. There are two major limitations on the size of a portable terminal: the power supply (batteries) and the I/O system (usually liquid crystal displays and push buttons). If the I/O system can be made invisible, great progress toward tiny, portable terminals will be possible.

Certainly, the wrist terminal cannot be far away. There are already portable terminals under development which can interface with local radio systems for hospital and industrial work. The portable, wrist-sized interactive terminal quickly can become a reality in industry and medicine.

Another major feature of desk-sized terminals will be extensive color-graphics capabilities. Graphics systems will use digitized pictures of very high quality for a variety of purposes.

MESSAGE SYSTEMS

The trend in message systems will be for more features and interconnection. Practical programs are under development that will allow one system to automatically dial another and transfer messages addressed to users of the called system. This would be the beginning of a true store-and-forward message service, which could beat the postal service in speed and cost (at low volume).

Higher-speed transmission (1200 baud) using the Bell 212 modem standard will become common. The large memory-storage capabilities of hard disks and video disks will allow the use of programming languages and data storage rivalling the present information utilities. Practical multi-user access will also become common in systems serving large user populations.

INFORMATION UTILITIES AND NETWORKS

Information utilities and value-added carriers will begin to look even more alike as they take on each other's roles. Value-added carriers are already offering such features as electronic mail service to their customers. Information utilities will take advantage of satellite transmission capabilities to extend their services throughout the country and, therefore, become their own carriers.

Information utilities of all types will continue to proliferate. Corporations will begin to offer access to information utilities to all executives—not just to receive information, but also to conduct business.

Information utilities will continue to expand their on-line data bases. The available data bases will rival community libraries in research material and excel any other source of current information. They will offer such services as digitized voice and video. Voice synthesis terminals will tie to information utilities to read aloud the data presented on the screen.

A Beginning

These predictions, like this book, are meant to stimulate you to explore the world of data communication. Data communication systems can allow your mind to travel out in the world and explore new areas of knowledge and current events in a unique and interactive fashion. Your microcomputer can provide the key to the door of knowledge and adventure. Come join us out on the data network.

Appendix

Anderson Jacobson
521 Charcot Avenue
San Jose, California 95131
Terminals, printers, and modems

Apparat, Inc.: Uniterm/80, TRS-80 Hardware and Software
4401 S. Tamarac Parkway
Denver, Colorado 80237
(800) 525-7674

Bill Blue
P.O. Box 1318
Lakeside, California 92040
(714) 449-4222
Peoples' Message System software

CompuServe
Personal Computing Division
5000 Arlington Centre Boulevard
Columbus, Ohio 43220
(614) 457-8600
    Information Utility

CP/M Users Group
1651 Third Avenue
New York, New York 10028
    CP/M software

Emtrol Systems, Inc.
123 Locust Street
Lancaster, Pennsylvania 17602
(717) 291-1116
    LYNX

GTE Telenet
8330 Old Courthouse Road
Vienna, Virginia 22180
(703) 827-9200
    Value-added carrier

Hazeltine Corporation
10 East 53rd Street
New York, New York 10022
    Terminals

Hayes Microcomputer Products, Inc.
5835 Peachtree Corners East
Norcross, Georgia 30092
    Integrated modems

Lance Micklus, Inc.
217 South Union Street
Burlington, Vermont 05401
    ST80 Communication Software

Lear Siegler, Inc.
714 North Brookhurst Street
Anaheim, California 92803
    Terminals

Lifeboat Associates
1651 Third Avenue
New York, New York 10028
CP/M Operating System and programs

Lindbergh Systems
41 Fairhill Road
Holden, Massachusetts 01520
(617) 852-0233
OMNITERM

MCI Communications Corp.
1150 17th Street N.W.
Washington, D.C. 20036
(202) 872-1600
Alternative telephone service

Microperipheral Corporation
P.O. Box 529
Mercer Island, Washington, 98040
(206) 454-3303
Microconnection & software

Microstuf, Inc.
1900 Leland Drive
Suite 12
Marietta, GA 30067
(404) 952-0267
Software for North Star and CP/M systems

MSI Data Corporation
340 Fischer Avenue
Costa Mesa, California 92626
(714) 549-6000
Portable terminals

NIXDORF Computer
168 Middlesex Turnpike
Burlington, Massachusetts 01803
(800) 225-1992
Portable terminals

North Star Computers, Inc.
1440 Fourth Street
Berkeley, California 94710
Microcomputers

NOVATION
18664 Oxnard Street
Tarzana, California 91356
Modems

Perkin-Elmer
360 Route 206 South
Flanders, New Jersey 07836
Terminals

Pickles & Trout
P.O. Box 1206
Goleta, California 93017
(805) 967-9563
TRS-80 Model II CP/M

RCA Microcomputer Products
New Holland Ave.
Lancaster Pennsylvania 17604
VP 3300 Color Terminals

Small Systems Business Group
6 Carlisle Rd.
Westford, Massachusetts 01886
(617) 692-3800
ST80 Communications Software

Software Sorcery, Inc.
7927 Jones Branch Drive Suite 400
McLean, Virginia 22102
(703) 385-2944
ABBS Software

Southwestern Data System
10159-G Mission Gorge Road
Santee, California 92071
(714) 562-3670
Communications software

SP Communications
P.O. Box 974
Burlingame, California 94010
(415) 692-5600
Alternative telephone service

TYMNET
20665 Valley Green Drive
Cupertino, California 95014
(408) 446-7000
Value-added carrier

Glossary

ABBS. Apple Bulletin Board System. An electronic message system based on the Apple II computer. Software written by Bill Blue and Craig Vaughan.

ACK. A positive acknowledgment control character.

Acoustic coupler. The portion of a modem which physically holds a telephone handset in two rubber cups.

ANSI. American National Standards Institute. Accepts standards for codes, alphabets, and signalling schemes.

ASCII. American Standard Code for Information Interchange. More correctly known as USASCII because of some changes to the most recent versions. A method of coding digital signals. The ASCII character contains 7 bits; an eighth parity bit is often added.

Asynchronous. A method of transmission in which the characters are not required to be in perfect timing. Start and stop bits are added to coordinate the transfer of characters.

Baud. A measure of transmission speed. The reciprocal of the time duration of the shortest signal element in a transmission. In RS-232-C ASCII transmission the signal element is 1 bit.

Baudot code. A coding scheme for data transmission using a 5-bit (5-level) code. Still used in communication systems for the deaf. The code now known as "Baudot" was really written by Donald Murray and replaced the older Baudot scheme. It is now sometimes known as Murray code. CCITT alphabet number 2.

BCD. A coding scheme using a 6-bit (6-level) code.

Bell 103. Modem protocol using four tones for full-duplex operation on a single channel. Usually considered to be limited to 300 baud.

Bell 113. Modem protocol identical to Bell 103 above.

Bell 202. Modem protocol using two tones for half-duplex transmission. Maximum speed on dialed circuits is 1200 baud. The mark tone = 1200 Hz and the space tone = 2200 Hz.

Bell 212. A dual-mode modem protocol featuring full-duplex transmission of speeds up to 300 baud using the 103 protocol or up to 1200 baud using a phase-shifted carrier. Not compatible with 202-type devices.

Bit. Smallest unit of information. In digital signalling commonly refers to a change in state between 0 and 1.

BPS. Bits Per Second.

Block. A number of characters transmitted as a group.

Buffer. A temporary storage space, usually in RAM.

Byte. A group of 8 bits.

CBBS. Computer Bulletin Board System. An electronic message system based on S-100 bus hardware and the CP/M operating system. Developed by Ward Christensen and Randy Suess.

CCITT. Consultive Committee on International Telegraph and Telephone. A committee of the United Nations which accepts international standards.

Character. One letter, number, or special code.

Common carrier. Transmission companies (telephone, etc.) which serve the general public.

Control character. A character used for special signalling. Often not printed or displayed, but causing special functions such as screen clear, printer tab, and others.

CPU. Central Processing Unit.

CPS. Characters Per Second.

Current loop. An electrical interface that is sensitive to current changes as opposed to voltage swings. Often used with older teleprinter equipment.

Cursor. The point of light indicating the place on a video screen where the next character will appear.

DAA. Data access arrangement. A device used to connect to and isolate telephone lines.

DB-25. The designation of the standard plug and jack set used in RS-232-C wiring. Twenty-five pin connectors with thirteen pins on the top and twelve pins on the bottom row.

DCE. Data Communications Equipment. A common designation for communication equipment such as computers and modems. Uses a female DB-25 chassis jack.

DTE. Data Terminal Equipment. A common designation for communications equipment such as printers and terminals. Uses a male DB-25 chassis plug.

EBCDIC. Extended Binary Coded Decimal Interchange Code. An 8-bit code used primarily on IBM business systems.

Echo-plex. A method of transmission in which characters are echoed from the distant end, and the echoes are presented on the terminal.

Echo suppressor. A device used to eliminate the echo effect of long-distance, voice-transmission circuits. These suppressors must be disabled for full-duplex data transmission. The modem answer tones turn the suppressors off automatically.

EIA. Electronic Industries Association.

FCC. Federal Communications Commission.

Forum 80. An electronic message system based on the TRS-80 hardware. Software written by Bill Abney.

FSK. Frequency Shift Keying. A transmission method using two different frequencies–like a modem.

Full-duplex. The ability to talk both ways at the same time.

Half-duplex. Alternating transmissions–like CB (over).

Handshaking. Exchange of control codes or specific characters to control the data flow.

Interface. The interconnection point–usually between equipment.
I/O. Acronym for "Input/Output."

K. Abbreviation for kilo, a prefix meaning 1000; for example, an 8K memory has approximately 8000 bytes of storage.

KSR. Keyboard Send and Receive. A terminal with a keyboard.

Mark. A signalling condition equal to a binary 1.

Message switching. A switching technique using a message store-and-forward system.

Modem. Modulator and Demodulator. A device which translates between electrical signals and audio tones. The audio tones may be used over telephone or radio circuits.

NAK. A control code indicating a character or block of data was not received.

Network. A communication system made up of various stations. The phrase *network* assumes interaction between the stations.

On-line. Connected to a network or host computer system.

Parallel transmission. Simultaneous transmission of all of the bits in a byte down 8 parallel wires.

Parity. A check of total number of 1 bits in a character. In ASCII, a final 8th bit is set so the count (when transmitted) is either al-

ways even or always odd. This even or odd state can be easily checked at the receiving end.

PMS. Peoples' Message System. An electronic message system based on Apple II hardware. Software written by Bill Blue.

Protocol. A set of rules governing the transmission of information over a data channel.

RAM. Random access memory. The workspace memory in a computer that can be written over, erased, and used again.

Reverse channel. An "answer back" channel provided during half-duplex operation. Allows the receiving modem to send low-speed acknowledgments to the transmitting modem without breaking the half-duplex mode.

ROM. Read Only Memory.

RS-232-C. An electrical standard for the interconnection of equipment established by the Electronic Industries Association. Practically identical to CCITT recommendation V.24.

SNA. Systems Network Architecture. A local networking scheme developed by IBM.

Space. The signal condition that equals a binary zero.

ST80. A series of communication programs written by Lance Micklus.

Start bit. A data bit used in asynchronous transmission to signal the beginning of a character.

Stop bit. A data bit used in asynchronous transmission to signal the end of a character and an idle channel. A mark condition lasting longer than the normal data bit.

Store and forward. A system, usually used in message switching, where messages are held until the appropriate receiving party is available.

Synchronous. A transmission system in which characters are synchronized by the transmission of initial sync characters. No stop or start bits are used.

WPM. Words Per Minute.

Index

A

PERSONAL COMPUTERS

TRS-80 DATA COMMUNICATION SYSTEMS

Here's a comprehensive computer handbook that shows how any student, businessperson, or hobbyist can use the TRS-80 to communicate with information utilities and message systems.

Presented in an easily understood fashion with little technical jargon, **TRS-80 DATA COMMUNICATION SYSTEMS** describes commercially available hardware and software combinations that can give the TRS-80 very powerful data communication capabilities.

Coverage includes:
• information utilities such as Source and CompuServe
• electronic message systems such as Forum-80 and CBBS
• how to share information and programs over telephone lines
• information on TRS-80 Model I, II, III, and Color Computer systems including operation of the Model II under CP/M
• how to select equipment that will extend the computer's capabilities • how to provide news, electronic mail, and financial services to a person with a data communication terminal • and much more.

FRANK J. DERFLER, JR., a telecommunications manager for the federal government, has been actively involved in the development of microcomputer systems and their application to data communication since 1977. He is the author of Microcomputer Data Communication Systems (Prentice-Hall) and currently writes a monthly column about microcomputer-based data communication systems for Kilobaud Microcomputing magazine.

A GUIDE TO THE OPERATION OF TRS-80 MICROCOMPUTERS AS COMMUNICATION DEVICES BY FRANK J. DERFLER, JR.

PRENTICE-HALL, Inc., Englewood Cliffs, New Jersey 07632
Cover design by Jeannette Jacobs

ISBN 0-13-931220-X